NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CONVERSATION THREAD EXTRACTION AND TOPIC DETECTION IN TEXT-BASED CHAT

by

Paige Holland Adams

September 2008

Thesis Advisor:	Craig H. Martell
Second Reader:	Cynthia E. Irvine

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*
26-09-2008

2. REPORT TYPE
Master's Thesis

3. DATES COVERED *(From — To)*
05-07-2006—26-09-2008

4. TITLE AND SUBTITLE
Conversation Thread Extraction and Topic Detection in Text-based Chat

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Paige Holland Adams

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of the Navy

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Text-based chat systems are widely used within the Department of Defense, but the standard systems available do not provide robust capabilities for search, information retrieval, or information assurance. The objective of this research is to explore methods for the extraction of conversation threads from text-based chat systems in order to enable such tasks. As part of the research, we manually annotated over 20,000 Internet Relay Chat posts with conversation thread information and constructed a probabilistic model for automatically classifying posts according to conversation thread. We also provide an algorithm for extracting these conversation threads from the chat session in order to form discrete documents that may be used in a vector space model information retrieval system. We elaborate how this technique can be used to support search and data mining systems, as well as auditing tasks and guard functions in a security system. Using the developed probabilistic models, we have achieved classification results on par with those of human annotators.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
Unclassified	Unclassified	Unclassified	UU	191	19b. TELEPHONE NUMBER *(include area code)*

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18

THIS PAGE INTENTIONALLY LEFT BLANK

CONVERSATION THREAD EXTRACTION AND TOPIC DETECTION IN TEXT-BASED CHAT

Paige Holland Adams
Lieutenant, United States Navy
B.S.B.A., Hawaii Pacific University, 2000

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2008

Author:	Paige Holland Adams
Approved by:	Craig H. Martell Thesis Advisor
	Cynthia E. Irvine Second Reader
	Peter J. Denning Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Text-based chat systems are widely used within the Department of Defense, but the standard systems available do not provide robust capabilities for search, information retrieval, or information assurance. The objective of this research is to explore methods for the extraction of conversation threads from text-based chat systems in order to enable such tasks. As part of the research, we manually annotated over 20,000 Internet Relay Chat posts with conversation thread information and constructed a probabilistic model for automatically classifying posts according to conversation thread. We also provide an algorithm for extracting these conversation threads from the chat session in order to form discrete documents that may be used in a vector space model information retrieval system. We elaborate how this technique can be used to support search and data mining systems, as well as auditing tasks and guard functions in a security system. Using the developed probabilistic models, we have achieved classification results on par with those of human annotators.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

1 INTRODUCTION **1**
1.1 MOTIVATION 1
1.2 ORGANIZATION OF THESIS. 2

2 BACKGROUND **3**
2.1 MILITARY CHAT REQUIREMENTS AND APPLICATION. 3
2.2 NATURAL LANGUAGE PROCESSING AND CHAT 6
2.3 CHAT FEATURES 12
2.4 INFORMATION RETRIEVAL. 24
2.5 TEXT CLASSIFICATION 26

3 TECHNICAL APPROACH **37**
3.1 DATA SETS 37
3.2 ANNOTATION 40
3.3 FIRST PHASE EXPERIMENTAL TECHNIQUES 41
3.4 SECOND PHASE EXPERIMENTAL TECHNIQUES 45

4 RESULTS **51**
4.1 ANNOTATOR OBSERVATIONS 51
4.2 INTER-ANNOTATOR AGREEMENT. 52
4.3 TIME-DISTANCE PENALIZATION RESULTS 54
4.4 MAXIMUM ENTROPY CLASSIFICATION RESULTS 58

5 CONCLUSIONS AND FUTURE WORK **65**

A ACRONYMS AND ABBREVIATIONS **67**

B GLOSSARY **69**

C CHAT ABBREVIATIONS AND INITIALISMS **73**

D COMMONLY ENCOUNTERED CHAT EMOTICONS **77**

E INTER-ANNOTATOR AGREEMENT RESULTS **83**
E.1 ##iphone METRICS 83
E.2 ##physics METRICS 85
E.3 #python METRICS 88

F MAXIMUM ENTROPY CLASSIFICATION RESULTS **93**
F.1 ##iphone, SAME ANNOTATOR, DIFFERENT SESSION 93
F.2 ##iphone, SAME SESSION, DIFFERENT ANNOTATOR 102
F.3 ##physics, SAME ANNOTATOR, DIFFERENT SESSION 104
F.4 ##physics, SAME SESSION, DIFFERENT ANNOTATOR 113
F.5 #python, SAME ANNOTATOR, DIFFERENT SESSION 115
F.6 #python, SAME SESSION, DIFFERENT ANNOTATOR 124

G MAXIMUM ENTROPY WITH LDA CLASSIFICATION RESULTS **127**
G.1 ##iphone, SAME ANNOTATOR, DIFFERENT SESSION 127
G.2 ##iphone, SAME SESSION, DIFFERENT ANNOTATOR 135
G.3 ##physics, SAME ANNOTATOR, DIFFERENT SESSION 138
G.4 ##physics, SAME SESSION, DIFFERENT ANNOTATOR 146
G.5 #python, SAME ANNOTATOR, DIFFERENT SESSION 149
G.6 #python, SAME SESSION, DIFFERENT ANNOTATOR 157

H CHAT TOOLS PYTHON CODE **161**

LIST OF REFERENCES **171**

INITIAL DISTRIBUTION LIST **173**

List of Figures

2.1 Calculate Message Dependency for Message i 10

2.2 Turn-taking Conversation Systems Array 15

2.3 Typical Chat Session 16

2.4 Illustration of Conversation Extraction Task. 22

2.5 One-to-one Annotation Metric 24

2.6 Local Agreement Metric 25

2.7 Sample Density Distribution over Simplex 30

2.8 LDA Replicant Plates 31

2.9 Data Model Comparison – Unigram, Unigram Mixture, pLSI 32

2.10 Topic and Word Simplex 33

2.11 AP Topic Category Example 34

2.12 Perxplexity Comparison over AP Corpus 35

2.13 Sample Topic Hierarchy Estimated Using hLDA. 36

3.1 Freenode IRC Server Room List 41

3.2 Chat Viewer Interface Showing Highlighted Threads 42

3.3 Connectivity Matrix Illustration 43

3.4 Thread Extraction Algorithm 45

3.5 Maximum entropy classifier 46

3.6 Maximum Entropy Classifier with LDA Topic Selection 49

4.1 Number of Speaker Posts vs. Number of Conversation Threads 52

4.2 Comparison of Thread Detection Techniques Across Threshold Values for Selected Thread of Interest 56

4.3 Effect of Time-Distance Penalization on Chat Post Association 57

4.4 Comparison of Classification Results across All Sessions - Maximum Entropy Classification 61

4.5 Comparison of Classification Results across All Sessions - Maximum Entropy Classification with LDA 63

List of Tables

2.1 US Navy Text-Based Chat Usage in Pacific Fleet and Indian Ocean Areas . . 4
2.2 COMPACFLT Mission Areas in Which Chat Is Used 5
2.3 Consolidated Functional Requirements for Tactical Military Chat 7
2.4 42 Dialog Act Labels for Conversational Speech 9
2.5 27 Initial Post Features 11
2.6 15 Post Act Classifications for Chat 12
2.7 12 Dialog Act Classifications for Task-Oriented Instant Messaging 12
2.8 Turn Allocation Techniques in Spoken Language 14
2.9 Chat Session Extract Illustrating Use of Mentions 18

3.1 Sample Topic Banner from #python Channel 38
3.2 Conversation Thread-annotated Files from Freenode IRC Server 40
3.3 Thread Detection Techniques 44

4.1 Summary Annotation Metrics for ##iphone Chat Sessions 53
4.2 Summary Annotation Metrics for ##physics Chat Sessions 53
4.3 Summary Annotation Metrics for #python Chat Sessions 53
4.4 Maximum Entropy Results, Same Annotator, Different Sessions, ##iphone . . 59
4.5 Maximum Entropy Results, Same Session, Different Annotators, ##iphone . . 59
4.6 Maximum Entropy Results, Same Annotator, Different Sessions, ##physics . 59
4.7 Maximum Entropy Results, Same Session, Different Annotator, ##physics . . 60
4.8 Maximum Entropy Results, Same Annotator, Different Sessions, #python . . 60
4.9 Maximum Entropy Results, Same Session, Different Annotator, ##physics . . 60
4.10 Maximum Entropy Model versus Human Annotation Accuracy 60

4.11 Classification results of same-session training and testing (different annotators) across all sessions using maximum entropy classification. 61

4.12 Classification results of same-session training and testing (different annotators) across all sessions using maximum entropy classification with LDA topic detection. 62

Acknowledgements

There were many people responsible for the completion of this thesis and to whom I owe my gratitude. First and foremost, I would like to thank my advisor, Prof. Craig Martell, for giving me the opportunity to pursue incredibly interesting work, the freedom to explore different avenues, and the encouragement and support necessary in an undertaking of this magnitude.

I would additionally like to thank Prof. Kevin Squire for providing me with the knowledge, tools, and guidance to start unraveling the mysteries of machine learning and Prof. Cynthia Irvine for the support and motivation for this research and for introducing me to the surprisingly fascinating theoretical aspects of computer security and information assurance.

Thanks to Lt. Col. Eric Forsyth and Ms. Jane Lin, my predecessors in this line of research and without whom my task would have been significantly more difficult (if not impossible). Also thank you to my lab partner, Ensign Grant Gehrke, who entertained my strange ideas and had an uncanny knack for suggesting the correct course of action whenever I found myself stymied by a thorny problem.

A special thank you to my fearless annotators – Stephen Dimitriou, Eric Johnson, and Lawrence Keener – who slogged undauntedly through many hours of chat. Without their support this project would have never left the ground. Thanks also to Constantine Perepelitsa for providing superior technical support and so much more.

Thank you to Dr. LorRaine Duffy and Ms. Emily Medina of Naval Space and Warfare Systems Command San Diego (SSC-SD), whose enthusiastic embrace of chat in the Department of Defense inspired this research.

Acknowledgements to SSC-SD for providing the financial support for the equipment and travel used in this thesis.

Thanks to Donald Knuth and Leslie Lamport for TeX and LaTeX, respectively, which made the writing of this thesis more enjoyable than it would have been otherwise. And thanks particularly to Prof. Simson Garfinkel for the excellent NPS thesis template and an always rapid response to my countless requests for assistance.

Thanks to my parents for the guidance and support they have given me over the years. Thanks especially to my father, a U.S. Army National Guard veteran, who encouraged me to join the Navy.

Finally, none of this would have been possible without the tremendous support of my wife, Masako, whose patience knows no bounds, and my two wonderful daughters, Chelsea and Abigail. Rest assured, they deserve as much credit for this work as I.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
INTRODUCTION

1.1 MOTIVATION

In the last decade, computer-mediated communications (CMC) such as e-mail, chat, and instant messaging have transformed global information flow. In the US military, applications such as e-mail and chat have expanded beyond their use as administrative support tools and now function as warfighting enablers, enhancing and in some circumstances, supplanting, traditional tactical systems. Rapid communication has often played a decisive role in warfare and is an especially critical element in today's complex combat environment, where participants may be dispersed over great geographical distances, may have varying clearance levels or varying levels of "need-to-know," or may consist of multinational coalition partners. Tactical chat, in particular, has emerged as an indispensable tool for military professionals to communicate, analyze, and fuse information with peers and allies in a real-time environment.

Despite the numerous advantages, there are several challenges in realizing the full potential of computer-mediated communications. One such challenge, exacerbated by the proliferation in use of these tools, is how to find and extract useful information information rapidly. This is a particularly difficult task in media such as chat due to the highly dynamic conversational environment coupled with a typically large number of participants. Another significant challenge is in the bridging of these applications across domain boundaries, whether from an SI to a GENSER network, or between US and coalition partner systems. The risk of losing tactical advantage due the time delay required for an air gap transfer of information to take place is real. This delay can be minimized through the use of guards that connect systems with different trust levels and allow the exchange of authorized data. Existing guards use techniques such as labeling and keyword filtering to manage secure information flow; however, these mechanisms are not able to detect knowledge inference within message content, therefore the possibility of sensitive information "leakage" remains.

To increase the value of tactical chat to the warfighter, we wish to address these two main challenges, namely: 1) information retrieval and 2) information filtering. This thesis presents an overview of chat and current state-of-the-art natural language processing techniques and related work that may be employed to help in achieving our goals. We then present a methodology and

algorithms for processing chat, along an evaluation of the results.

1.2 ORGANIZATION OF THESIS

We have organized this thesis as follows. In Chapter 1 we provide the motivation for chat analysis and the development of techniques for information extraction and filtering. Chapter 2 provides: 1) an overview of chat, including its linguistic structure and comparison with other forms of dialog in spoken and written communications, 2) an overview of the tactical chat requirements, and 3) general natural language processing techniques as well as related NLP chat work. In Chapter 3 we detail our technical approach, to include a discussion of the chat corpora used, the algorithms employed, and the set-up of our experiments using this data along with the evaluation metrics. Chapter 4 discusses the results of our experiments, specifically the performance of our algorithms on the following three tasks: 1) conversation thread extraction, 2) topic detection and retrieval, and 3) topic filtering. In Chapter 5 we conclude with a summary of our work along with recommendations for future research.

CHAPTER 2: BACKGROUND

In this chapter we briefly discuss the requirements for military use of tactical chat, then examine areas where natural language processing (NLP) can support these requirements. We provide a background on commonly-used NLP techniques that address some of the tasks required, along with some statistical techniques that could be employed to augment performance. Finally, we discuss related work in the field and how some of these approaches might be used to address the concerns of tactical chat. Technical terms, acronyms, and abbreviations are provided for reference in Appendices B and A.

Fundamentally, the first task that we are interested in accomplishing is that of *information retrieval* (IR). Manning *et al.* define information retrieval as "finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)" [1, p. 1]. As this indicates, most IR tasks involve searching across discrete collections of documents, *e.g.,* text documents in an file system or web pages on the Internet. With chat, however, the IR task is slightly more complex. With standard search tools one could search across a collection of archived chat logs and return those that match based upon the search criteria. A problem with this approach is that the file may be quite large and contain a large volume of posts by many participants. These posts may comprise many conversations about a great number of topics. The searcher is likely only interested in a single topic or smaller subset of topics. The ideal scenario would be to return only the topic-related posts and, for contextual purposes, other posts in the same conversation thread. This is the task that we set out to accomplish in this study. Before addressing the specifics of how that task might be accomplished, we feel that it is instrumental to first look at how chat is currently being used in the military and to what degree.

2.1 MILITARY CHAT REQUIREMENTS AND APPLICATION

Text-based chat is used extensively by all military branches and throughout the Department of Defense. It is used for unit-level tactical coordination as well as broad-scale strategic planning and joint operations. Increasingly, it is becoming a preferred tool for communication between disparate platforms or with coalition partners. In 1996, Eovito conducted a comprehensive

PRNOC	
Area	Pacific Fleet
Servers	2 primary, 1 backup
Chat rooms	400-500 (typical), 500-650 (exercise)
Users	400-600 (typical), 600-100 (exercise)

IORNOC	
Area	Indian Ocean and Arabian Gulf
Servers	1 primary, 1 backup
Chat rooms	500-650 (typical)
Users	900-1300 (typical), 5000+ (major combat operations)

Table 2.1: US Navy text-based chat usage in Pacific Fleet and Indian Ocean areas

survey of joint tactical chat usage [2], which provides a useful starting point for our discussion.

2.1.1 Fleet Tactical Use

In [2], Eovito outlined requirements for a joint tactical chat system based upon a study of actual chat usage in several different environments: combat operations in Operation ENDURING FREEDOM, counter-insurgency operations in Operation IRAQI FREEDOM, and disaster relief operations in support of Joint Task Force - Katrina. Eovito notes that the use of chat among joint forces has evolved in an ad hoc fashion in an effort to fill gaps in existing command and control (C2) systems, but has become an essential communications tool favored over more traditional methods. The aim in this study was to determine actual operator requirements based upon the capabilities and usage of current chat systems so that these requirements can be used in the development of future C2 systems.

In a 2008 survey conducted by the Naval Space and Warfare Systems Command [3], US Navy Fleet commands were asked questions regarding their text-based chat usage, including specific mission areas in which it was used as well as number of servers and users. Chat server usage as reported by the Pacific Regional Network Operations Center (which overs the Pacific Fleet area of operations) and the Indian Ocean Regional Network Operations Center (whose responsibility includes the Indian Ocean and Arabian Gulf) are found in Table 2.1. Some of the mission functions in which chat plays a role, as reported by COMPACFLT, are in Table 2.2.

Chat, as a command and control medium, has several advantages over other C2 systems, particularly in a naval environment. Some of the advantages outlined in Eovito's study include:

Mission Area
Over-the-horizon targeting coordination
Intelligence
Information warfare command
Link coordination
Logistics
Maritime interdiction operations
Tomahawk land attack missile coordination
Maritime security operations
Anti-terrorism/Force protection coordination
Combat cargo operations
Air resource element coordination
Meteorological weather coordination
Medical coordination
Mine warfare operations
Coast Guard/Homeland security
Marine Forces intelligence collaboration
Training

Table 2.2: COMPACFLT mission areas in which chat is used.

1. *Bandwidth*. The bandwidth requirements for text-based chat are far less than for other data systems. This is important in bandwidth-constrained tactical environments, particularly for smaller naval tactical units which have less available bandwidth.

2. *Speed*. Chat is faster than other systems both due to rapid transmission time of text and also due to the more rapid turnaround as compared to other methods such as message traffic, or even radio or phone calls since chat provides for simultaneous transcription and dissemination.

3. *Ease-of-use*. Most chat clients have a very shallow learning curve compared to other C2 systems, requiring less training.

4. *Availability*. Users typically experience a higher degree of availibility of chat compared to other C2 systems. According to [2], users "reported that chat was the only form of communication in many cases, where units were too far for voice, and the available transmission systems lacked the bandwidth for larger C2 systems." Also, many Command, Control, Communications, Computer, and Intelligence (C4I) plans call for chat to be one of the first systems available when deployed, making it useful as a coordination tool for bringing other C2 systems online.

5. *Efficiency*. Tactical users often find that "chat allows them to send more data with less time and effort" [2]. Also, it is easy to monitor chat while working with other onscreen tools, maps, etc. Since chat provides a running transcript, users spend less time having to repeat information that was previously disseminated, and as they may participate in multiple chat rooms, it is easier to target a designated audience.

Based on current chat usage patterns coupled with existing C2 requirements, Eovito suggests requirements for future tactical chat systems (see Table 2.3). Both CENTCOM and NORTHCOM have cross domain requirements for chat, with CENTCOM's requirements stating that a system should be "capable of sending messages between different networks of various security [classifications]." This implies a need for ensuring that the messages sent do not violate security policies in the process.

2.1.2 Data Mining

Eovito's thesis concludes by listing several areas for future research in support of tactical chat. One such area is data mining. According to Eovito, "[m]odern data and text mining tools applied to chat logs present unique knowledge discovery opportunities" [2]. It is the aim of this thesis to take steps in that direction and explore the structure of chat and how we might exploit features inherent in chat to enable data mining systems.

2.1.3 Information Assurance

With the desire to use chat as a bridge across multi-domain environments comes an even greater need for attention to information assurance implications. Accordingly, we also examine topic management within the context of information assurance, *i.e.,* we attempt to provide methods for auditing chat sessions to locate topics that may have security considerations, as well as discuss possibilities for online chat guards that can allow or disallow topics consistent with a defined security policy.

2.2 NATURAL LANGUAGE PROCESSING AND CHAT

Statistical natural language processing (NLP) techniques are frequently employed in the analysis and processing of spoken conversation. These tools and methods that NLP provide have recently proven useful in the analysis of text-based chat as well. In this section, we provide an overview of relevant NLP methodology and its application toward chat analysis. In particular,

1. Participate in Multiple Concurrent Chat Sessions*
2. Display Each Chat Session as a Separate Window
3. Persistent Rooms and Transitory Rooms*
4. Room Access Configurable by Users
5. Automatic Reconnect and Rejoin Rooms*
6. Thread Population/Repopulation*
7. Private Chat “Whisper”*
8. One-to-One IM (P2P)
9. Off-line messaging
10. User Configured System Alerts
11. Suppress System Event Messages
12. Text Copying*
13. Text Entering*
14. Text Display*
15. Text Retention in Workspace*
16. Hyperlinks
17. Foreign Language Text Translation
18. File Transfer
19. Portal Capable
20. Web Client
21. Presence Awareness/Active Directory*
22. Naming Conventions Identify Functional Position*
23. Multiple Naming Conventions*
24. Multiple User Types
25. Distribution Group Mgmt. System for Users
26. Date/Time Stamp*
27. Chat Logging*
28. User Access to Chat Logs*
29. Interrupt Sessions
(* denotes a core requirement)

Table 2.3: Consolidated functional requirements for tactical military chat (from [2])

we begin with a discussion of recent NLP work involving chat, then discuss several statistical NLP techniques that may be applied to chat.

2.2.1 Author Profiling

Detecting sexual predator and other illegal activity within chat is a common goal since the medium has a strong attraction for individuals with this type of behavior. Toward this end, automatic author profiling – determining the gender, age, background, etc., of an author – is desired in order to determine, for example, if someone is attempting to hide his or her true identity. Lin conducted a study of techniques for author profiling within a chat domain [4] in

which approximately 400,000 posts from age-specific chat rooms were collected and analyzed. This chat currently forms the core of the NPS Chat Corpus (a more complete discussion of which is found in Chapter III), which was one of the key corpora used in our research.

Lin selected surface details of the collected chat conversations to include average number of words per post, size of the vocabulary, use of emoticons, and the use of punctuation [4]. Using the author's self-reported profile to establish the "true" age and gender, Lin then used the naïve Bayes method to classify each user based upon these features. Although this initial study had mixed results, it highlighted several areas for future improvement, including the usage of a more comprehensive surface feature set such as distribution over all words, and the inclusion of deeper features (*e.g.,* syntactic structure).

In order to enable further methods such as those proposed by Lin, Forsyth developed a richer NLP chat methodology [5]. Taking advantage of Lin's work, he sought to lay the groundwork for further analysis of the syntactic structure of chat through the automatic tagging of part-of-speech and dialog act information.

2.2.2 Dialog Act Modeling

A dialog act is the description of the role that a given sentence, phrase, or utterance plays in a conversation. For example, *Is it raining today?* would be labeled as a **YES/NO Question** to indicate the role that it plays in the conversation, which also serves as an indication of its relationship with other posts in the same conversation thread. Labeling of dialog acts is typically conducted manually, but can be a tedious task. Several studies have been conducted on building probabilistic models for automatic dialog act labeling.

In [6], Stolcke *et al.* describe a method for the automatic dialog act labeling of utterances in conversational speech by treating the discourse structure of a conversation as a hidden Markov model. Training and evaluating the model using 1,155 conversations drawn from the Switchboard corpus of spontaneous human-to-human conversational speech, they achieved a model accuracy of 65 percent based on automatic word recognition and 71 percent based on word transcripts. This compares to a human accuracy of 84 percent on the same task. The 42 dialog acts found within Switchboard along with an example and their frequency of occurrence in the database are shown in Table 2.4.

Forsyth [5] applied a modification of techniques described in [6] to text-based chat. Using the

Tag	Example	Percent of Total
Statement	Me, I'm in the legal department.	36%
Backchannel/Acknowledge	Uh-huh.	19%
Opinion	I think it's great.	13%
Abandoned/Uninterpretable	So, -/	6%
Agreement/Accept	That's exactly it.	5%
Appreciation	I can imagine.	2%
Yes-No-Question	Do you have to have any special training?	2%
Non-Verbal	<Laughter>, <Throat_clearing>	2%
Yes Answers	Yes.	1%
Conventional-Closing	Well, it's been nice talking to you.	1%
Wh-Question	What did you wear to work today?	1%
No Answers	No.	1%
Response Acknowledgment	Oh, okay.	1%
Hedge	I don't know if I'm making any sense or not.	1%
Declarative Yes-No-Question	So you can afford to get a house?	1%
Other	Well give me a break, you know.	1%
Backchannel-Question	Is that right?	1%
Quotation	You can't be pregnant and have cats.	0.5%
Summarize/Reformulate	Oh, you mean you switched schools for the kids.	0.5%
Affirmative Non-Yes Answers	It is.	0.4%
Action-Directive	Why don't you go first.	0.4%
Collaborative Completion	Who aren't contributing.	0.4%
Repeat-Phrase	Oh, fajitas.	0.3%
Open-Question	How about you?	0.3%
Rhetorical-Questions	Who would steal a newspaper?	0.2%
Hold Before Answer/Agreement	I'm drawing a blank.	0.3%
Reject	Well, no.	0.2%
Negative Non-No Answers	Uh, not a whole lot.	0.1%
Signal-Non-Understanding	Excuse me?	0.1%
Other Answers	I don't know.	0.1%
Conventional Opening	How are you?	0.1%
Or-Clause	or is it more of a company?	0.1%
Dispreferred Answers	Well, not so much that.	0.1%
3rd-Party-Talk	My goodness, Diane, get down from there.	0.1%
Offers, Options, & Commits	I'll have to check that out.	0.1%
Self-talk	What the word I'm looking for	0.1%
Downplayer	That's all right.	0.1%
Maybe/Accept-Part	Something like that.	$< 0.1\%$
Tag-Question	Right?	$< 0.1\%$
Declarative Wh-Question	You are what kind of buff?	$< 0.1\%$
Apology	I'm sorry.	$< 0.1\%$
Thanking	Hey, thanks a lot	$< 0.1\%$

Table 2.4: 42 dialog act labels for conversational speech. (From [6]) Percentage indicates the frequency of posts in the corpus with the given dialog act label.

NPS Chat Corpus, Forsyth successfully automated part-of-speech tagging of chat posts with a 90.8 percent accuracy and dialog act classification with a 83.2 percent accuracy. For dialog act classification, Forsyth used a set of fifteen classification labels constructed by Wu *et al.*

$d \leftarrow i$
repeat
$d \leftarrow d - 1$
$typing\_rate \Leftarrow \frac{time(M_i) - time(M_d)}{/} length(M_i)$
until $typing\_rate < typing\_threshold$ or $d = 1$ or $speaker(M_i) = speaker(M_d)$

Figure 2.1: Calculate message dependency for message i (from [8])

[7] specifically for text-based chat dialog. These labels are shown in Table 2.6. The best-performing dialog act classification model was constructed by using a neural network with 23 input features. The complete set of 27 features tested by Forsyth are shown in Table 2.5.

For the POS-tagging task, Forsyth evaluated several tagging methods including using n-gram taggers, hidden Markov model (HMM) taggers, and Brill transformational-based learning taggers trained on a variety of sources which included the Wall Street Journal, Brown corpus, Switchboard, Penn Treebank, and others. The best performance in this study was realized by a tagger that used combination of techniques: the Brill tagger, with back off to the HMM, and n-gram taggers. This approach achieved a mean accuracy of 90.8 percent. This was followed by the HMM tagger with a mean accuracy of 88.5 percent [5].

Another approach to dialog act tagging, using instant messaging (IM) instead of chat, was undertaken by Ivanovic [8]. This work was aimed at an analysis of online shopping assistance provided by the MSN Shopping website. Ivanovic's approach differed from that of the Wu and Forsyth studies in that he considered the dialog act of utterances in the conversation stream independent of the post level. An utterance under this scheme can span more than one post or contain multiple utterances in a single post. Ivanovic's initial task of utterance segmentation was accomplished manually by hand-annotation of the dialog acts within each post using the twelve dialog act labels show in Table 2.7. Ivanovic then applied an algorithm (shown in Figure 2.1) to re-synchronize the posts in order to overcome the inherent asynchrony of the message stream. This algorithm used typing rate and time between posts to determine, given a pair of posts, whether one post was dependent upon the other. Dependency in this case was defined in terms of a message being posted by a user having had knowledge of the preceding post. The second post would then be deemed as dependent upon the first. Using these resynchronized threads with a naïve Bayes classifier and an n-gram model (n = 1, 2, and 3), Ivanovic achieved an average bigram (units of evaluation comprising two words) accuracy of 81.6 percent.

Feature	Definition	Rationale
f0	Number of posts ago the poster last posted	Indicator for a Continuer act
f1	Number of posts ago the poster made a spelling error	Indicator for a Clarify act
f2	Number of posts ago that a post contained a '?' but no WRB or WP POS tag	Indicator for a Yes/No Answer act
f3	Number of posts in the future that contained a Yes or No word	Indicator for a Yes/No Question act
f4	Number of posts ago that contained a Greet word	Indicator for a Greet act
f5	Number of posts in the future that contained a Greet word	Indicator for a Greet act
f6	Number of posts ago that contained a Bye word	Indicator for a Bye act
f7	Number of posts in the future that contained a Bye word	Indicator for a Bye act
f8	Number of posts ago that a post was a JOIN	Indicator for a Greet act
f9	Number of posts in the future that a post is a PART	Indicator for a Bye act
f10	Total number of words in post	Longer posts may be Statements and Questions, shorter posts may be Emotions and Greets/Byes, etc.
f11	First word is a conjunction, preposition, or ellipses (POS tag of 'CC,' 'IN,' or ':')	Indicator for a continuer act
f12	A word contains emotion variants such as 'lol,' ';-),' etc.	Indicator for an emotion act
f13	A word contains 'hello' or variants	Indicator for a Greet act
f14	A word contains 'goodbye' or variants	Indicator for a Bye act
f15	A word contains 'yes' or variants	Indicator for Yes or Accept acts
f16	A word contains 'no' or variants	Indicator for No or Reject acts
f17	A word POS tag is 'WRB' or 'WP'	Indicator for a Wh-Question act
f18	A word contains one or more '?'	Indicator for Wh- or Yes/No Question acts
f19	A word contains one or more '!' (but not a '?')	Indicator for an Emphasis act
f20	A word POS tag is 'X'	Indicator for an Other act
f21	A word is a system command ('.' or '!' with SYM POS tag)	Indicator for a System act
f22	A word is a system word, *e.g.,* JOIN, MODE, ACTION, etc.	Indicator for a System act
f23	A word is an 'any' variant, *e.g.,* 'anyone,' 'n e,' etc.	Indicator for a Yes/No Question act
f24	A word is in all caps, but not a system word like 'JOIN'	Indicator for an Emphasis act
f25	A word is an 'even' or 'mean' variant	Indicator for a Clarify act
f26	Total number of users currently in the chat room	More users may stretch out distances between adjacency pairs

Table 2.5: 27 initial post features (from [5])

Tag	Example	Percent
Statement	I'll check after class	42.5%
Accept	I agree	10.0%
System	Tom [JADV@11.22.33.44] has left #sacbal	9.8%
Yes-No-Question	Are you still there?	8.0%
Other	**********	6.7%
Wh-Question	Where are you?	5.6%
Greet	Hi, Tom	5.1%
Bye	See you later	3.6%
Emotion	lol	3.3%
Yes-Answer	Yes, I am.	1.7%
Emphasis	I do believe he is right.	1.5%
No Answer	No, I'm not.	0.9%
Reject	I don't think so.	0.6%
Continuer	And. . .	0.4%
Clarify	Wrong spelling	0.3%

Table 2.6: 15 post act classifications for chat (from [7])

Tag	Example	Percent
Statement	I am sending you the page now	36.0%
Thanking	Thank you for contacting us	14.7%
Yes-No-Question	Did you receive the page?	13.9%
Response-Ack	Sure	7.2%
Request	Please let me know how I can assist	5.9%
Open-Question	how do I use the international version?	5.3%
Yes-Answer	yes, yeah	5.1%
Conventional-Closing	Bye Bye	2.9%
No-Answer	no, nope	2.5%
Conventional-Opening	Hello Customer	2.3%
Expressive	haha, :-), grr	2.3%
Downplayer	my pleasure	1.9%

Table 2.7: 12 dialog act classifications for task-oriented instant messaging (from [8])

2.3 CHAT FEATURES

In order to perform tasks such as classification on chat, we must first identify features which may inform our classification model. A useful starting point in feature identification is to look at the basic characteristics of that which we are trying to classify. Much work has been done in the examination of the dynamics of spoken conversation, so we will begin with an overview of general conversation characteristics, then turn toward those features that distinguish text-based chat from spoken conversation.

2.3.1 Conversation Features

As defined by Zitzen and Stein, "[c]hat programs are multi-user, synchronous, computer-mediated communications systems, which allow communication among spatially distal participants" [9].

In its basic form, chat is most similar to spoken conversation, sharing many characteristics with multi-party spoken dialog. Thus, it is useful to examine the dynamics of spoken conversation as a starting point for our chat analysis. In particular, we are interested in turn-taking and what factors influence this in spoken dialog as well as chat. Sacks *et al.* [10] noted the following basic observations regarding spoken conversation:

- Speaker-change recurs, or at least occurs.
- Overwhelmingly, one party talks at a time.
- Occurrences of more than one speaker at a time are common, but brief
- Transitions (from one turn to a next) with no gap and no overlap are common. Together with transitions characterize by slight gap or slight overlap, they make up the vast majority of transitions.
- Turn order is not fixed, but varies.
- Length of conversation is not specified in advance.
- What parties say is not specified in advance.
- Number of parties can vary.
- Talk can be continuous or discontinuous.
- Turn-allocation techniques are obviously used. A current speaker may select a next speaker (as when he addresses a question to another party); or parties may self-select in starting to talk. See Table 2.8 for a full description of turn-allocation techniques.
- Various 'turn-constructional units' are employed; *e.g.,* turns can be projectedly 'one word long,' or they can be sentential in length.
- Repair mechanisms exist for dealing with turn-taking errors and violations; *e.g.,* if two parties find themselves talking at the same time on of them will stop prematurely, thus repairing the trouble.

1. The current speaker may implicitly or explicitly select the next speaker, who is then obliged to speak.
2. If the current speaker does not select the next speaker, the next speakership may be self-selected. The one who starts to talk first gets the floor.
3. If the current speaker does not select the next speaker, and no self-selected speakership takes place, the last speaker may continue.
4. If the last (current) speaker continues, rules 1-3 reapply. If the last (current) speaker does not continue, the the options recycle back to rule 2 until speaker change occurs.

Table 2.8: Turn allocation techniques in spoken language (from [10])

Aoki *et al.* detailed several qualitative phenomena of spoken conversation in a study of multi-party interaction [11]. They note the existence of floors – instantiations of the turn-taking mechanism in effect – and remark that it is not uncommon for multiple floors to exist within a social participation framework. They use Egbert's definition of schism as "the emergence of an additional floor amidst ongoing floor(s)" in a multi-party interaction [12]. Three phenomena that lead to schism were outlined:

1. Schism by Schism Inducing Turn. Described by Egbert as having three characteristics:
    - It causes a change in topic.
    - It is the first part of a pair of turns (such as the question in a question-answer pair) that initiates a new sequence.
    - It directly targets a specific recipient or recipients.
2. Schism by Toss-Out. A "toss-out" is defined as a type of action that is topic-relevant to the conversation at hand, does not target a specific audience, and does not require a response or acknowledgement. Aoki *et al.* observe three different outcomes that may result from a toss-out:
    - No response may be generated. No new conversation floor emerges.
    - A response may be generated that follows the trajectory of the in-process conversation. No new conversation floor emerges.
    - A response may be generated that creates a new trajectory parallel to the conversation that produced the toss-out. A new conversation floor is created.

3. Schism by Aside. Asides are similar to toss-outs in that they are topic-relevant to the ongoing conversation and they do not require a response. The biggest distinction between the two is that asides are designed to be intentionally marginal to the ongoing conversation. In spoken conversation, these may be differentiated audibly, for example, by speaking in a more subdued tone. In chat, an aside might be marked by text in parentheses or some other delimiter that sets it apart from the main utterance. A chat initialism, emoticon, or IRC action may also be an indicator for an aside.

Sacks describes differential turn-taking systems as scale with one polar extreme being represented by one-turn-at-a-time allocation instances such as face-to-face conversation and the other extreme by preallocated turn instances as typified by debates. Admitting text-based chat to this model, we might consider an extension to the scale with chat forming a new extreme opposite the preallocation pole and face-to-face conversation occupying a location in between these poles (see Figure 2.2). This array is representative of the flexibility of the turn-taking system being used.

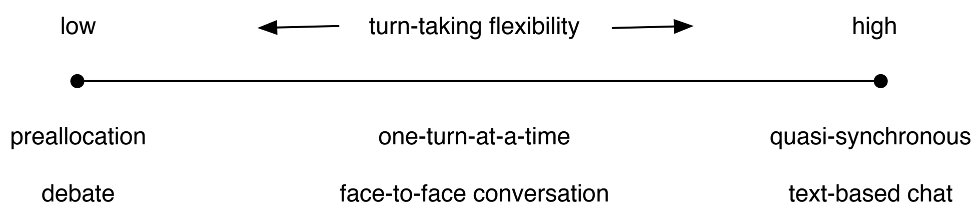


Figure 2.2: Turn-taking conversation systems array

To underscore the differences between chat and spoken conversations, Zitzen and Stein suggest that in chat “a much more intricate and complicated layering of partial [turn-taking] mechanisms” exists beyond those suggested by Sacks [9]. In particular, the role that technology plays is emphasized. For example, the speaker selection properties listed in Table 2.8 are replaced by a “first message to server, first message posted to dialog frame” method of conversation-floor selection. Thus, personal relationships perform a secondary role in selection for chat, rather than a primary role as in spoken conversation.

An additional difference noted by Zitzen and Stein involves the concepts of *hearer* and *speaker*. In spoken conversation, these roles are discrete and distinct; an individual can only perform in one role at a given time. In chat, however, the delineation is not as sharply drawn. A “hearer” may be “speaking” (*i.e.,* typing a response) at the same time that a message is received. Similarly, many individuals may be “speaking” (typing) at the same time. Which individuals holds

the floor is determined by which message arrives at the server first and either: 1) continues the conversation or 2) generates a schism.

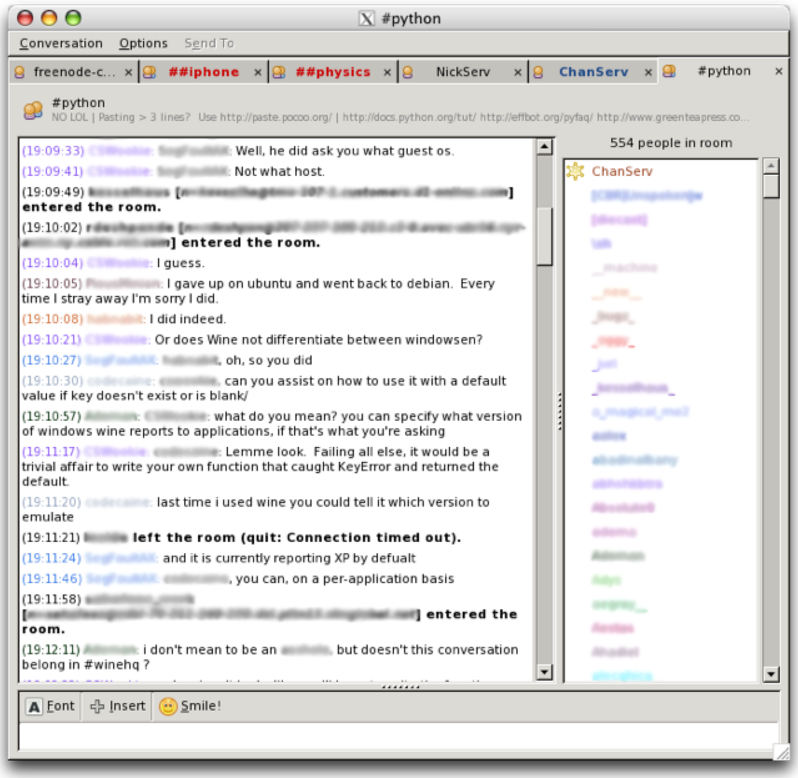


Figure 2.3: A typical chat session shown in pidgin chat client. (User names and identifying information intentionally blurred for anonymity.)

2.3.2 Chat Specific Features

Although chat is in many ways similar to spoken conversation, it does have characteristics which make it unique and which could serve as useful features in building a classification model for conversation thread detection. The following is a discussion of some the more important of these characteristics.

- Chat initialisms (CIs) are abbreviations and acronyms that have arisen in chat to convey common actions or commonly expressed emotions. For example, the phrase *be right back* is often abbreviated as *BRB* and *laughing out loud* (used to denote or convey appreciation of humor in a post or posts) becomes *LOL*. A more complete list of commonly used initialisms can be found in Appendix C.

- Emoticon usage. Emoticons are symbols formed from ASCII characters that express an emotion or mood and are often used as a proxy for speech or body language cues that are not available in text-based chat. A list of commonly-used emoticons can be found in Appendix D. An interesting point to note regarding emoticons is that, although they are used in many different cultures and languages, there is a distinct difference in style between Western emoticons and Eastern emoticons. Western-style emoticons are generally "read" by tilting one's head to the left, turning the horizontal ASCII characters into a vertical depiction of a character. Constrastingly, Eastern-style emoticons are typically designed to be read in a horizontal format. For example, a face may be formed by `(*_*)`, where the underscore represents a mouth and the asterisks form eyes. In Japan, such emoticons are known as *emoji* and are quite standardized in usage. It is common to find *emoji* character sets built into mobile phones for use in text messaging and mobile e-mail.

- Abbreviated speech (grammar/spelling shorthand) – misuse of grammar and spelling is often more tolerated in chat than in other forms of communication, and may in many cases be intentional.

- Mentions. In order to clarify to whom a particular post is directed, the technique of *mentioning* is often used. This most often takes the form of using the targeted user's name in a post, though it might also take the form of repeating a key word or words of the post or posts to which it is responding. An example of the use of mentions is shown in Table 2.9.

- Textual devices. Chat participants often use clever textual devices other than emoticons as a method of clarification or adding additional information. For example, if a mention is omitted from a response, the responder may immediately follow up with the user's name and a caret symbol ('`^`') to indicate that the preceding post being pointed to by the symbol is directed toward that user. Users also use this symbology self-referentially, posting some variation of '`<----`' to indicate that an action or statement refers to the user themself.

Antonietta	why does it tell me my list is a non-sequence now ...
Marcy	that's no list
Tanna	*Demarcus:* bar = (percent * '#') + ((percent - bar_size) * '-')
Demarcus	looks fine
Antonietta	*Marcy:* actually, it told me that when i did for (index, entry) in list: (i forgot enumerate) is that right ?
Tanna	test time
Tanna	hmm, instead of putting '-' s, it just leaves blank spaces, let me try something
Mickie	hey guys... in the try/except block... in the except block for the err... how can i capture the err/msg generated by the app when it fails..??
Marcy	*Antonietta:* it took an entry from the list, and tried to unpack it into the two variables
Tanna	oh, got it I guess: bar_size - percent, would be the right thing
Tanna	yeah that was it :)
Antonietta	*Marcy:* yes, i understand that it broke, but should it tell me its trying to iterate a non-sequence ?
Tanna	thanks *Demarcus*, looks much better now :)
Demarcus	np

Table 2.9: Chat session extract illustrating use of mentions (in italics).

2.3.3 Social Networking in Chat

The social nature of chat lends itself to an analysis of the network of relationships that are formed in the course of a chat session (and across multiple sessions). We are at the beginning stages of exploring the effect that user participation has on topic thread detection by considering user names ("nicknames") as a feature in our post vector. The intuition behind this is that, other considerations aside, a post by a given user is more likely to be associated with the conversation with which the user's previous post was associated.

Tuulos and Tirri [13] conducted a detailed analysis of the use of social network analysis and topic models in chat data mining. An observation made in their research was that, unlike in face-to-face conversation where non-verbal cues such as eye contact and physical proximity dictate the targeting of a conversation, chat must rely on verbal cues. This means, for example, that individual posts targeted toward a certain recipient will often contain the nickname of that recipient in the text of the post[1].

Tuulos and Tirri augmented chat topic models with social networking information using graph-based features such as the indegree, outdegree, and complementary outdegree of a node that represents a chat user. Additionally, they applied Google's PageRank concept to this graph-

[1]Some chat clients provide a convention for targeting posts toward a particular user. For example, to target a post to a particular user, that user's nickname is prepended with an "@" symbol. The nickname is then hyperlinked to that user's profile or message stream.

based model and experimented with filtering and biased sample weighting schemes. Their results showed that the indegree of a chat user node was the best indicator of a chat user with topic predictive content in their posts.

2.3.4 Interactional Coherence

Although conversation thread disentanglement is a difficult task for a computer, human beings can do it quite well. O'Neill and Martin analyzed human performance in tracking interaction in text based chat [14]. Their study refuted previous contentions that the unique properties of text-based chat, *e.g.,* quasi-synchronicity and potential for multiple simultaneous conversations, can lead to interactional incoherence. Their study is a useful starting point for examining the way human beings work together in a chat environment for constructing a coherent conversation. By looking at human methodology, we might discover methods useful in training a machine to accomplish a similar task.

Previous researchers cited a lack of control over turn positioning as one problem contributing to interactional incoherence in chat. That is, due to the simultaneity property, there is no guarantee that turns will appear in the order that would be expected in a face-to-face conversation. An answer to a question, for example, may not directly follow the question to which it is responding. There may in fact be several unrelated or partially-related posts in between. O'Neill and Martin note that other researchers of text-based chat have perceived this lack of serial adjacency to be a cause of thread confusion, since location of a turn in spoken conversation is partially responsible for being able to determine its meaning. They cited this concern as the impetus for a redesign of user interfaces in an attempt to compensate for the multi-threading. In these interfaces, users could select the thread to which their post belonged and the posts would appear spatially separated according to thread. A problem noted with this is that participants had no specific point of focus in the interface since new entries could appear anywhere in the chat space. This led, in fact, to more confusion as humans seem to have a cognitive preference for temporal ordering of conversation turns.

It was also suggested that the presence of "phantom" adjacency pairs was a source of incoherence. That is, the lack of serial adjacency of actual conversation pairs may lead users to perceive that an interleaving post is related to a preceding post, when it is in fact not.

O'Neill and Martin also cited studies that provided evidence contradicting the interactional incoherence theory. One such study by Herring [15] suggested that the features of chat (*e.g.,*

loose inter-turn connectedness and overlapping exchanges) alleged to attribute to the problem may in fact produce positive benefits, such as the ability for users to participate in multiple simultaneous conversations within a single discussion. This is something that is much more difficult to do in spoken conversation. A unique feature of chat that allows this to occur is its persistence, *i.e.,* the previous conversations stay on the screen, or can be easily scrolled to, so that they are available for reference.

Other chat features noted in research cited by O'Neill and Martin were that delays in response were not treated as noticeably absent as would be the case in spoken conversation. Also, in order to increase referent/message coherency, posters frequently post rapidly, using short utterances and splitting longer messages into smaller ones. Posters also make structural decisions, conscious or otherwise, to enable their audience's understanding of their message even in the event of interleaving. Mentions (which O'Neill and Martin refer to as "naming") and repetition are two common techniques used in this regard. Another feature noted was that it was rare for participants to use one turn to answer more than one previous turn – multiple response turns were preferred.

In their paper, O'Neill and Martin explain that "[m]ultiple threads can consist of parallel chats with different participants in each thread or participants may be involved in two threads simultaneously." Indeed, there is technical upper bound on the number of threads in which a user may participate; however, there may be very real limits on cognition and performance as thread participation increases. This is an interesting cognitive science question in its own right, but it is outside the scope of our objectives for this paper

O'Neill and Martin, in their own study, analyzed chat that was recorded during a a series of online business seminars. The participating audience was small (6 to 11 users), but were geographically dispersed in such locations as the UK, Russia, and Canada. The participants were professional business people, both acquainted and unacquainted, with varying levels of technical ability. The sessions analyzed were in the range of 60 to 90 minutes.

In their observations, O'Neill and Martin noted that the persistence aspect played a key role in multiple thread management. Even though chat scrolled out of the visible portion of the screen after a period of time, this did not prevent users from referencing these posts. According to their findings, "[p]articipants' entries during these events show that they do use this feature (for example, in [one event] one participant answered a much earlier query to him well after it would have been visible without scrolling.)" O'Neill and Martin also observed that

> most chat entries are easily associated with the thread to which they contribute because of the **observable contextual relations**. That is, the contributions in a thread are sequentially related to one another in an **accountable** way (*i.e.,* the relations are observable and reportable) even where their serial relations have been disrupted by intervening comments from different threads [14].

This statement suggests that indeed there are tangible features (observable contextual relations) that link related posts together. If true, these features might prove useful in building a model for machine learning.

O'Neill and Martin do not suggest that misunderstandings never occur in chat, but note that the turn-taking system anticipates this and makes allowances for the misunderstanding or confusion to be corrected in the following turn. As in the previous studies performed by other researchers, O'Neill and Martin also observed the use of mentions in chat to forestall possible confusion when the situation warranted. They noted that since conversation works "on the basis of economy," the explicit use of other users' names in the conversation performs as a "failsafe to ensure more conversational effort is not required in order to identify the desired recipient"[14].

Recent research closely aligned with the goals of our study is that of Elsner and Charniak in [16]. Their study presented a method for disentangling conversations from Internet Relay Chat (IRC) using a graph theoretic approach and maximum entropy classification. Elsner and Charniak define disentanglement as "the clustering task of dividing a transcript into a set of distinct conversations." The specific classification task is to decide, for each pair of posts in a given chat session, if the posts belong to the same conversation.

Figure 2.4 depicts the thread extraction task using one thread for purposes of illustration. In this case, the thread in question is a conversation regarding where a person lives in South Africa. The posts comprising this conversation are intermingled with other topics within the chat stream and, in fact, the participants in this thread may be simultaneously involved in other non-related conversations. What distinguishes this as a separate conversation is the dialog interaction between posts, the relative stability of participants, and the stability of the topic. Note however that these are not hard and fast rules: in chat, just as in spoken conversation, participants may enter and leave and the topics may shift or change altogether over time. The key factor is that when these events occur in a conversation thread, they typically do so with a noticeable transition phase rather than abruptly. For example, when new participants enter a conversation,

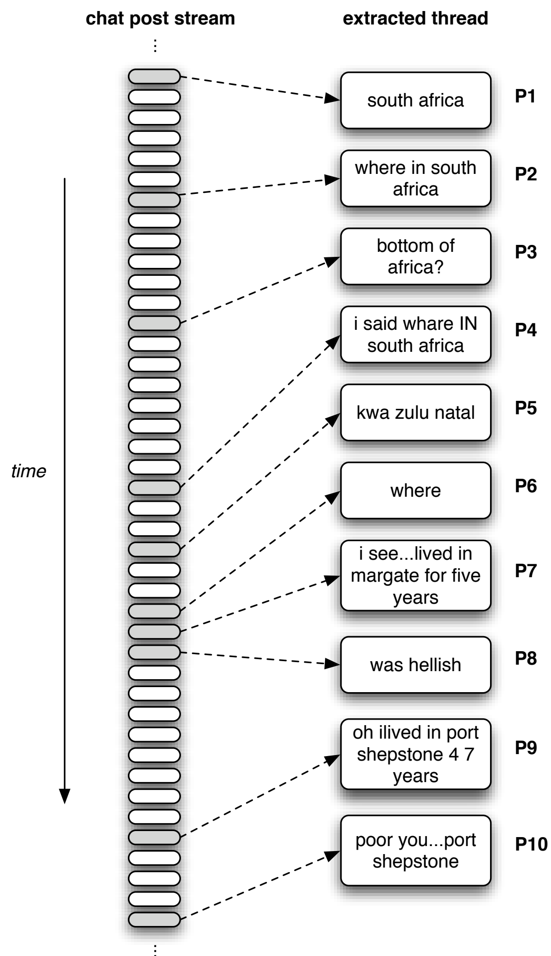


Figure 2.4: Illustration of conversation extraction task. Multiple conversations in a session are interleaved. The goal in extraction is to select only those posts that belong to a given conversation thread.

they will typically greet the existing particpants, who in turn will return the greeting. Likewise, departures are marked by farewells. Topic change is often a response to some stimulus in the conversation or will be explicitly marked by a partipant (*e.g., By the way* ... or *I hate to change the subject, but* ...).

Nigam *et al.* were among the first to explore using maximum entropy techniques for text classification in [17]. In this study, the goal was to compare the performance of maximum entropy classification against other supervised learning techniques, particularly naïve Bayes. This initial examination revealed that maximum entropy in some cases performed significantly better than naïve Bayes, but in other cases it performed worse. The study did, however, serve to show that maximum entropy can be effective in text classification and pointed out several areas in which the technique can be improved.

Nigam *et al.* explain that the concept behind maxium entropy is simply "that one should prefer the most uniform models that also satisfy any given constraints" [17]. To illustrate this concept the following example was offered:

> [C]onsider a four-way text classification task where we are told only that on average 40% of the documents with the word "professor" in them are in the **faculty** class. Intuitively, when given a document with "professor" in it, we would say it has a 40% chance of being a **faculty** document, and a 20% chance for each of the other three classes. If a document does not have "professor" we would guess the uniform class distribution, 25% each. This model is exactly the maximum entropy model that conforms to our known constraint [17].

The Elsner and Charniak maximum entropy classifier employs three different categories of features:

- Chat-specific. These features include time gap between posts, the speaker, and mentions.
- Discourse. Includes cue words (*e.g.,* "hello" to denote greeting), questions (marked by a question mark), and long posts (greater than 10 words).
- Content. Repeat(i) (words shared between two posts with unigram probability i, bucketed logarithmically), Technical (two posts use of technical jargon).

In order to provide a meaningful measure of the performance of a classification model, we must compare it to human performance on the same task. Therefore, it is important that we determine the level of agreement of multiple annotators on the same data. To evaluate inter-annotator agreement, as well as the performance of their maximum entropy classification model, Elsner and Charniak employed three different sets of evaluation methods:

- One-to-one accuracy - global accuracy that measured the total percentage overlap (see Figure 2.5)
- Local agreement - the percentage of agreements within some context k, where k is number of preceding utterances (see Figure 2.6)

- Many-to-one - comparative measure of detail in annotation; maps each conversation of source annotation to the single conversation in the target annotation with which it has greatest overlap, then counts total percentage of overlap.

The one-to-one accuracy and local agreement methods are evaluation methods are illustrated in Figures 2.5, 2.6.

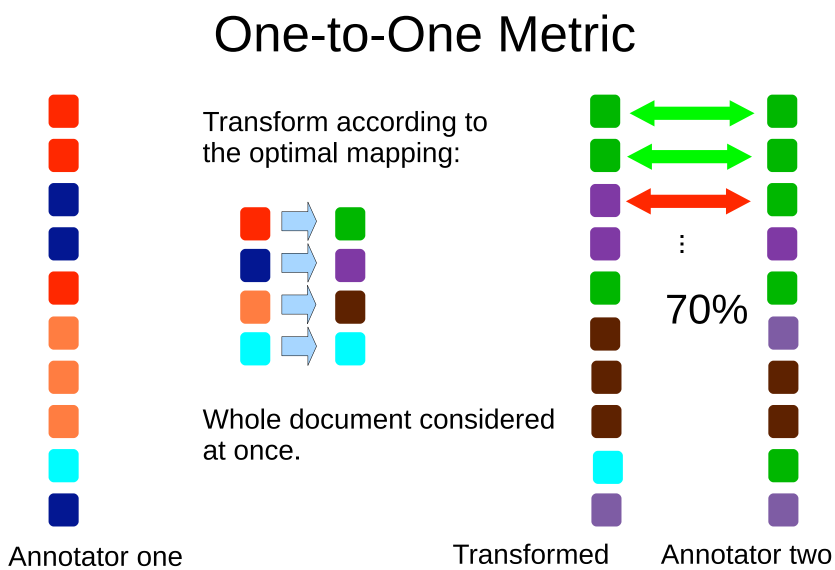


Figure 2.5: One-to-one annotation metric (from [18]).

2.4 INFORMATION RETRIEVAL

Once conversation threads are extracted from a chat session, we might treat these threads as distinct documents within a document space. The task then becomes one of search, *i.e.,* how to retrieve the conversations ("documents") in which we are interested. This is a well-studied field and many excellent methods exist for enabling seach. The following is a brief description of one of the more popular approaches.

2.4.1 Vector Space Model

Our research makes extensive use of the vector space model—one of the most frequently used techniques in information retrieval systems. This model, described by Salton in [19], represents documents and queries as vectors of features. Often, these features are the terms (*e.g.,* n-grams) that occur within the document collection, with the individual value of each feature representing the occurrence or non-occurrence of the term within the document that it represents. If there

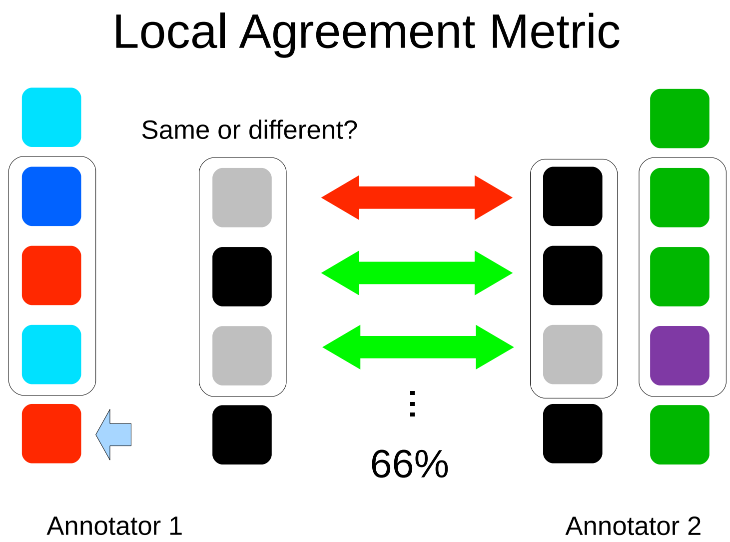


Figure 2.6: Local agreement metric (from [18]).

are N terms in a document collection, then each feature vector would correspondingly contain N dimensions.

In its simplest form, the feature value may use a binary value to indicate the existence of a term. A slightly more sophisticated model may incorporate the frequency of a term, under the presumption that the more often a term is used in a document, the greater the importance of that term to the document. This often has the unfortunate side effect of lending too much weight to common terms that may occur with a high degree of frequency throughout the entire collection, so schemes such as **term frequency-inverse document frequency** (**TF-IDF**) are used to discount these high frequency terms. Jurafsky and Martin [20] show a common formula for TF-IDF as

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{n_i}\right),$$

where the weight of a term i in the document vector for j is the product of its frequency, tf, in j and the log of its inverse document frequency in the collection, with n_i representing the number of documents in the collection that contain term i and N representing the total number of documents in the collection.

Our methodology makes use of this approach with the modification that, instead of documents, we are considering individual posts in a chat stream. Therefore, we utilize the frequency of a term in a post, discounted by the log of its inverse frequency across all posts in that stream.

Term frequency is but one of several term weighting schemes used. Other popular weightings include binary, logarithmic, and augmented normalized term frequency.

Finding similarity documents in a collection then becomes a matter of comparing vectors representing the documents and returning those that are "closer" in document space. Since the magnitude difference (due to relative term frequency) between vectors of documents with similar content could place the vectors further apart, the lengths of the vectors are typically normalized and proximity is based on cosine similarity as follows:

$$sim(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{||\vec{V}(d_1)||\ ||\vec{V}(d_2)||}$$

The numerator represents the dot product, or cosine similarity, of the vectors representing documents d_1 and d_2. The denominator is the product of the Euclidean lengths of the vectors, which serves to normalize the magnitudes.

Finding similarity between a query and a document in a collection is accomplished in like manner by performing comparisons between document vectors and a vector comprising the terms of the query.

2.4.2 Vector Space Model Usage

It is significant to note that we use the vector space model in two separate areas in our research: 1) TF-IDF is used in the time-distance penalization experiments detailed in Chapters 3 and 4 in order to established a weighting between posts, and 2) once the conversation threads are extracted, the vector space model is used to retrieve conversations of interest based on a search query. In fact, any search methodology may be used to accomplish the second task, and though the performance of the information retrieval task was not a part of this study, it is likely that are algorithms which may be particularly suitable for this.

2.5 TEXT CLASSIFICATION

As mentioned previously in the discussion of maximum entropy classification, text classification is the task of categorizing units of text (*e.g.,* words, sentences, paragraphs, or documents) based upon features of the text itself. In additon to maximum entropy, there are several classification techniques that are know to perform text classification well. This section contains a

description of three popular techniques. We include this discussion due to our choice to evaluate one of these – Latent Dirichlet Allocation (LDA) – in conjunction with the Elsner and Charniak maximum entropy classifier to improve the chat feature set.

In particular, one of the weaker features employed by the Elsner and Charniak classifier is the absence or presence of "technical" words in a post. This is based on the assumption that technical words are descriptive of a topic of interest. In the case of the Elsner and Charniak study, their corpus consisted entirely of chat from a single Linux-related session. Therefore, the assumption was that these technical words were descriptive of Linux-related topics. To generate the technical words list, Elsner and Charniak used a Linux technical manual and filtered out all words that were contained in a general news corpus (with news items pre-dating the Linux operating system), leaving only Linux-specific technical terms behind. This approach, while effective on the particular session used in the study, has several limitations:

1. Finding good source texts upon which to use the word-differential approach many be problematic.
2. This approach may not work with chat session that are not technical in nature.
3. Topics may in fact include non-technical words.
4. It does not account for multiple topics within the context of a global topic-oriented session.
5. It is difficult to update the model with additional information.

To address these limitations, we evaluated the use of LDA in constructing our feature set. The technical details and the results of this are included in Chapters 3 and 4.

2.5.1 Probabilistic Latent Semantic Indexing

Probabilistic Latent Semantic Indexing (pLSI; also known as Probabilistic Latent Semantic Analysis or pLSA) is a generative model for text classification proposed by Hofmann [21] that models in each word in a document as a sample from a mixture model. In pLSI each word is generated from a single topic; different words in the document may be generated from different documents. The output of this model is a list of mixing proportions for the different mixture components.

The pLSI model (see Figure 2.9(c)), proposes that a document label d and a word w_n are conditionally independent given an unobserved topic z:

$$p(d, w_n) = p(d) \sum_z p(w_n|z)p(z|d)$$

Although this model captures the possibility that a document may contain multiple topics given that $p(z|d)$ forms the mixture weights of the topics for a particular document d, a drawback to this approach is that d is simply an index into documents in the training set. This being the case, there is no natural way to assign a probability to a previously unseen document. Latent Dirichlet Allocation, which we now turn to, is an attempt to overcome this limitation.

2.5.2 Latent Dirichlet Allocation

Blei *et al.* describe Latent Dirichlet allocation (LDA) as "a generative probabilistic model for collections of discrete data such as text corpora" [22]. It is an approach similar to, and often compared with, the pLSI model. LDA is a three-level Bayesian model that assumes that items in a collection, such as documents when used in the context of text corpora, are formed as a finite mixture over a set of latent topics. These topics themselves are selected from an infinite distribution of topic probabilities. These topic probabilities predicted by the model form an explicit representation of a document. Although LDA is quite suited toward working with text, it has also proved beneficial in other patterned-data domains such as imaging and bioinformatics.

LDA aims to address some of the shortcomings of the pLSI model. Chiefly, as Blei *et al.* explain is that pLSI "provides no probabilistic model at the level of documents" [22]. The output of the pLSI model is a list of numbers that represent the mixing proportions for documents, but there is no generative model provided for the numbers. Two additional problems noted with this approach were that the model input parameters grow linearly with the size of the corpus, and there is not clear method for assigning probabilities to documents not contained within the training set.

The LDA model leverages the Dirichlet process introduced by [23], the formal definition of which is as follows:

Let Θ be a measurable space, with H a probability measure on the space, and let α be a positive real number. A *Dirichlet Process* is the distribution of a random probability measure G over Θ

such that, for any finite partition $(A_1, \ldots, A_r)$ of Ω, the random vector $(G(A_1), \ldots, G(A_r))$ is distributed as a finite-dimensional Dirichlet distribution:

$$(G(A_1), \ldots, G(A_r)) \sim \text{Dir}(\alpha H(A_1), \alpha H(A_r))$$

As explained by Blei *et al.* , the following generative process for each document w in a corpus D is assumed by LDA:

1. Choose word length $N \sim \text{Poisson}(\xi)$.
2. Choose topic mixture $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $Z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose a word w_n from $p(w_n|z_n, \beta)$, a multinomial conditioned on the topic z_n.

Some simplifying assumptions are in effect for this model: 1) the dimensionality k of the Dirichlet distribution is assumed known and fixed, and 2) the word probabilites are parameterized by a $k \times V$ matrix β, where $\beta_{ij} = p(w^j = 1|z^i = 1)$, which is initially treated as a fixed quantity to be estimated. The Poisson distribution over document length is also an assumption and one that is not critical to the Dirichlet process, therefore a more realistic distribution for document length may be substitued as desired.

A k-dimensional Dirichlet random variable θ can take values in the $(k-1)$-simplex and has the following probability density on the simplex:

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1},$$

where the parameter α is a k-vector with components $\alpha_i > 0$, and where $\Gamma(x)$ is the Gamma function. Figure 2.7 illustrates an example probability density on a two-dimensional simplex for distributions over three words and four topics.

Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics $\mathbf{z}$, and a set of N words $\mathbf{w}$ is given by:

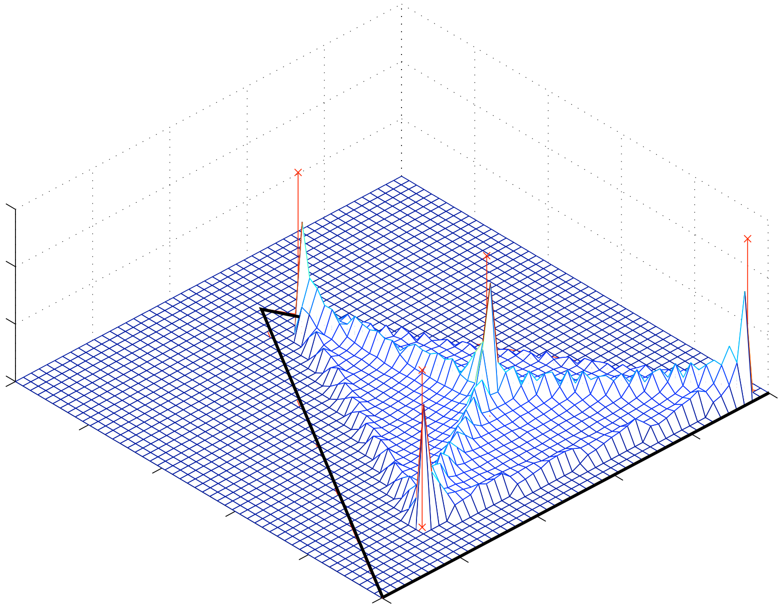


Figure 2.7: Example density on unigram distributions $p(w|\theta, \beta)$ under LDA for three words and four topics. The triangle shown on the plane is the two-dimensional simplex that represents all possible distributions over three words. (From [22].)

$$p(\theta, \mathbf{z}, \mathbf{w}|\alpha, \beta) = p(\theta|\alpha) \prod_{n=1}^{N} p(z_n|\theta) p(w_n|z_n, \beta),$$

where $p(z_n|\theta)$ is simply θ_i for the unique i such that $z_n^i = 1$. Integrating over θ and summing over z gives us the marginal distribution of a document:

$$p(\mathbf{w}|\alpha, \beta) = \int p(\theta|\alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n|\theta) p(w_n|z_n, \beta) \right) d\theta.$$

The last step is to take the product of the marginal probabilities of the single documents, giving us the probability of a corpus:

$$p(\mathbf{D}|\alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d|\alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn}|\theta_d) p(w_{dn}|z_{dn}, \beta) \right) d\theta_d.$$

As Blei *et al.* note, the parameters α and β are corpus-level parameters and are assumed to be sampled once in the process of generating a corpus. The θ_d variables are document-level and are sampled once per document. The z_{dn} and w_{dn} variables are at the word-level and are sampled once for each word in the document [22]. A graphical depiction of the LDA model illustrating these relationships is shown in Figure 2.8.

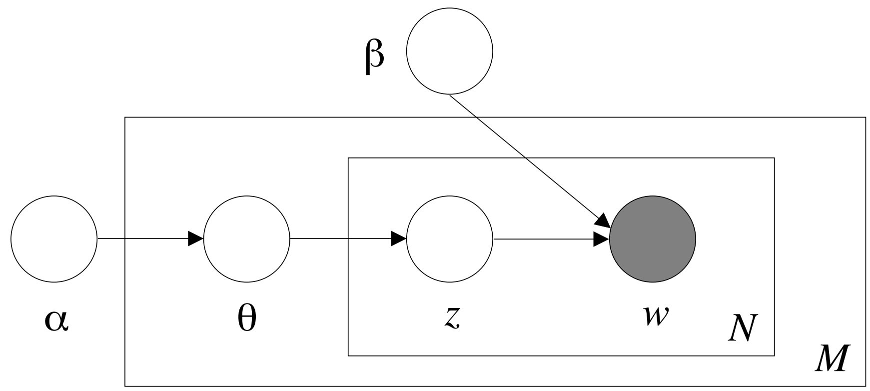


Figure 2.8: The boxes in the illustration of the LDA model indicate "plates" representing replicants. The outer plate is the replicant for documents, while the inner plates is the repeated choice of topics and words within the document. (From [22].)

The relationship of LDA with simpler latent variable models for text is described by Blei *et al.* [22]. Figure 2.9 shows a comparison of three different probabilistic models of discrete data: unigram, mixture of unigrams, and the pLSI/aspect model. Note the difference between these and the LDA model shown in Figure 2.8.

In the unigram model, illustrated in Figure 2.9(a), the words of every document are drawn independently from a multinomial distribution:

$$p(\mathbf{w}) = \prod_{n=1}^{N} p(w_n).$$

The mixture of unigrams model (Figure 2.9(b) is generated by augmenting the unigram model with a discrete random topic variable z. Documents are generated in this model by first selecting a topic z and generating N words independently from the conditional polynomial $p(w|z)$. The document probability is:

$$p(\mathbf{w}) = \sum_z \prod_{n=1}^{N} p(w_n|z).$$

According to Blei *et al.* [22], the mixture of unigrams model makes the assumption that each document represents exactly one topic. LDA, in contrast, allows documents to exhibit multiple topics to different degrees through the addition of one additional parameter. In the mixture of unigrams model, there are $k-1$ parameters associated with $p(z)$; whereas in LDA $p(\theta|\alpha)$ takes k parameters.

As discussed in the previous section, and provided here again for reference, the pLSI model assumes conditional independence of a document label d and a word w_n, given an unobserved topic z:

$$p(d, w_n) = p(d) \sum_z p(w_n|z)p(z|d)$$

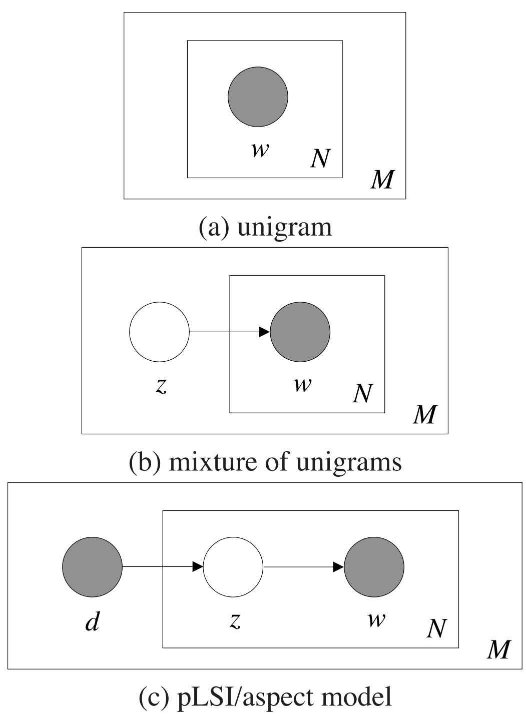


Figure 2.9: Graphical model representation of different models of discrete data. (From [22].)

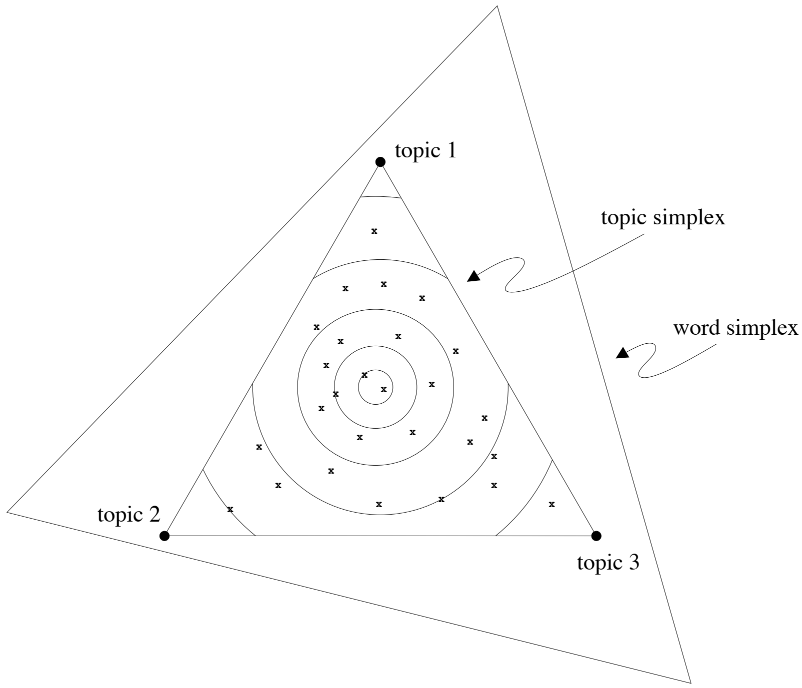


Figure 2.10: This figure shows the topic simplex for three topics embedded in the word simplex for three words. The corners of the word simplex represent the distribution where each word has a probablity of one. The topic simplex, likewise, has points that represent three different distributions over words that each correspond to a document (as a mixture of unigrams). In the pLSI model, an empirical distribution (denoted by the small x marks in this figure) is induced on the topic simplex. The LDA model places a smooth distribution (denoted by the contour lines) on the topic simplex. (From [22].)

In an evaluation of real-world performance, Blei *et al.* trained the LDA model on a subset of 16,000 documents from the TREC AP corpus. A 100-topic model was assumed and expectation maximization was used to find the Dirichlet and conditional multinomial parameters. Figure 2.11 illustrates some of the most probable words from several topics, which were then manually labeled with a representative tag. As can be seen, the LDA model is able to capture topical groupings that correspond to human intuition.

To evaluate generalization performance, Blei *et al.* compared LDA with the unigram, unigram mixture, and pLSI models. The models were trained on two text corpora containing unlabeled documents with the goal of achieving high likliehood on a held-out test set (90 percent training; 10 percent holdout). Perplexity, a measure often used in language modeling, was used as the metric for evaluation. Perplexity is monotonically decreasing in the liklihood of the test data and is algebraically equivalent to the inverse of the geometric mean per-word likliehood, with lower score indicating better performance. The formal definition of perplexity given a test set

"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.

Figure 2.11: Example article from the Associated Press corpus (from [22]). The color coding indicates the topic category from which the word was putatively generated.

of M documents is:

$$perplexity(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{M}\log p(\mathbf{w_d})}{\sum_{d=1}^{M} N_d}\right\}$$

The generalization performance of the four classification models is shown in Figure 2.12. The most important thing to note is effect that unseen documents have on the perplexity. An unseen document may best fit one of the components for the mixture models (mixture of unigrams or pLSI) but it will likely contain at least one word that did not occur in the training documents. These unseen words will, as a result, have a very small probability, causing the perplexity for the new document to increase dramatically. This is not the case for LDA, which consistently

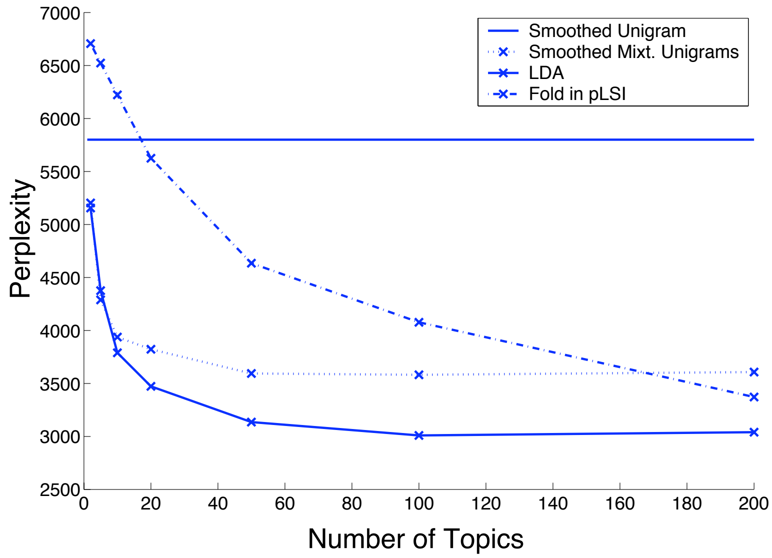


Figure 2.12: Perplexity results on the Associated Press corpus for LDA, the unigram model, mixture of unigrams, and pLSI. (From [22].)

outperformed the other classification models.

2.5.3 Hierarchical Latent Dirichlet Allocation

Though not evaluated in this study, a refinement of LDA known as hierarchical LDA (hLDA), uses a statistical sampling technique known as the Chinese Restaurant Process (CRP) in conjunction with the LDA approach. The advantage offered by hLDA is that it performs well when the number of topics in the distribution are not known beforehand, or an estimation of the number of topics is not feasible.

Figure 2.13 illustrates the performance of hLDA against a text data set of 1717 NIPS extracts. This corpus contained 208,896 words and a vocubulary of 1600 terms. From this Blei *et al.* [24] used hLDA to estimate a three-level hierarchy. The first level of the hierarchy consists of function words captured by the model. Because these types of words are not usually useful in distinguishing text for classification, they are often manually removed from a corpus prior to the learning process. This step is unnecesary in hLDA, as the system was able to detect these words automatically. In the second level of the hierarchy are words assocated with the topic categories of neuroscience and machine learning. Finally, the third-level hierachy contains words associated with important subtopics in these categories.

Having completed this overview of chat and related natural language processing work, we will

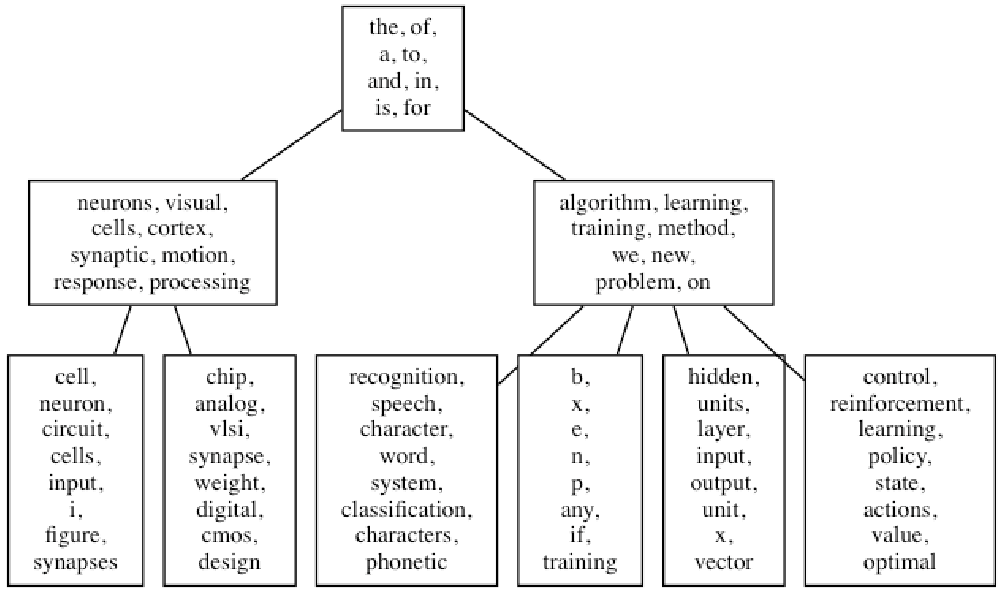


Figure 2.13: Sample topic hierarchy estimated from 1717 abstracts from NIPS01 through NIPS12 using hLDA (from [24])

now turn to the technical details of our research.

CHAPTER 3:
TECHNICAL APPROACH

3.1 DATA SETS

In our study, we used two primary data sets: 1) the original NPS Chat Corpus and 2) new sessions collected from IRC chat channels (which are being incorporated into the NPS Chat Corpus). A full description of both data sets is as follows:

3.1.1 NPS Chat Corpus

The NPS Chat Corpus, discussed in Chapter 2, was initially collected in 2006 by Lin [4]. Lin collected in excess of 475,000 chat posts by more than 3200 users from five different age-oriented rooms at (non-IRC) Internet chat site. The chat rooms were socially-oriented and not bound by specific topic, hence the discussions contained therein are diverse. This chat was subsequently POS and dialog act-tagged by Forsyth [5]. Currently 10,567 posts are tagged in this manner and are publicly available[1] in XML format.

Although this corpus has no time-stamp information associated with constituent posts, the ordering of posts in each session is preserved.

3.1.2 Freenode IRC

We augmented the original NPS Chat Corpus with additional chat collected from the Freenode IRC server during late July 2008. The motivation for this was to replicate tactical military chat as closely as possible in an unclassified environment to permit more freedom for annotation, analysis, and broader dissemination. Figure 3.1 shows a sample of chat rooms available on the Freenode IRC server. Chat sessions were recorded using the open source pidgin[2] client.

Collecting this chat provided us with two added advantages: 1) we were able to preserve time stamp information and 2) we were able to select topic-specific IRC channels. In all, over 504 minutes of chat from three separate channels were collected in this stage. Details of these chat sessions, including file name, number of non-system lines in file, and duration of each session in minutes, are show in Table 3.2. Channels were chosen based upon number of users and activity

[1] Available at `http://faculty.nps.edu/cmartell/NPSChat.htm`
[2] `http://www.pidgin.im/`

```
  NO LOL | Pasting > 3 lines? Use http://paste.pocoo.org/
| Tutorial: http://docs.python.org/tut/
| FAQ: http://effbot.org/pyfaq/
| New Programmer? Read
  http://www.greenteapress.com/thinkpython/
| #python.web #wsgi #python-fr #python.de #python-es
  #python.tw #python.pl #python-br
| IRSeekBot logs this channel publicly at
  http://www.irseek.com/
```

Table 3.1: #python channel policy as set forth in topic banner. This channel has an explicit "NO LOL" policy in an attempt to curtail needless banter (noise) in channel. This policy is routinely "enforced" by channel participants.

level, global topic, and low "noise" content (*e.g.,* the discussions tended to focus around the global topic of the chat room without excess social banter). The following is a description of each channel:

Channel Name	Description
#python	active channel devoted to Python programming, moderately high technical level with question-answer conversation
##physics	active channel with scientific (but not necessarily technical) conversation with sustained discussion threads
##iphone	active channel due to recent release of new Apple iPhone model); slightly "noisier"; more opinion-based conversation.

The low-noise aspect of the chosen channels can be attributed to three main factors: 1) the nature of the chat room global topic, 2) posted channel rules, and 3) enforcement by users. The "channel rules" refers to the text that is typically included in the topic banner for the chat room (set in IRC by issuing the `/topic` command)(see example in Table 3.1). This banner appears in room listings and is also displayed within the active chat channels. It often includes explicit rules for members to follow, typically to avoid a surfeit of off-topic banter. Users who break these rules risk sufferering criticism from other users and in the worst cases (and depending upon the level of moderation of the chat room by channel operators – those with elevated status in the room), may find themselves banned from the channel. As an example, in the course of one of the collected Python sessions, a participant used the 'lol' chat initialism in violation of the posted "NO LOL" policy. The user was chastised for this by the other chat participants, which induced a new conversation thread relating to the "NO LOL" policy.

File	Non-system lines	Duration
iphone_07_17.txt	251	3:38:13
iphone_07_18.txt	585	5:46:12
iphone_07_19.txt	748	45:58:07
iphone_07_21.txt	591	13:52:38
iphone_07_22.txt	844	13:44:07
iphone_07_23.txt	241	11:40:41
iphone_07_24.txt	831	15:10:51
iphone_07_25.txt	603	12:55:42
iphone_07_26.txt	392	11:54:56
iphone_07_27.txt	335	9:36:27
iphone_07_28.txt	242	9:28:40
iphone_07_29.txt	331	7:31:53
iphone_07_31.txt	110	10:04:55
Total	6104	171:23:22
physics_07_17.txt	67	3:29:55
physics_07_18.txt	99	5:48:11
physics_07_19.txt	438	45:56:58
physics_07_21.txt	702	13:55:22
physics_07_22.txt	203	13:35:54
physics_07_23.txt	137	11:38:22
physics_07_24.txt	703	15:04:45
physics_07_25.txt	750	13:01:19
physics_07_26.txt	828	12:00:12
physics_07_28.txt	487	9:21:59
physics_07_29.txt	504	7:40:34
physics_07_31.txt	120	9:54:53
Total	5038	161:28:24
python_07_17.txt	323	3:40:29
python_07_18.txt	716	5:48:47
python_07_19.txt	736	45:56:14

continued on following page. . .

File	Non-system lines	Duration
python_07_21.txt	706	13:55:22
python_07_22.txt	768	13:46:01
python_07_23.txt	735	11:39:47
python_07_24.txt	673	15:10:11
python_07_25.txt	670	13:02:21
python_07_26.txt	697	11:58:55
python_07_27.txt	775	9:41:14
python_07_28.txt	704	9:32:41
python_07_29.txt	597	7:42:44
python_07_31.txt	683	10:05:22
Total	8783	172:00:08
Combined Total	19925	504:51:54

Table 3.2: Conversation thread annotated chat files from Freenode IRC server. Duration given in HH:MM:SS.

3.2 ANNOTATION

The IRC chat was hand-annotated by conversation thread by three annotators comprising one college undergraduate and two high school interns. All possessed a basic understanding of the Python programming language and have taken physics-based classes, giving them some degree of background knowledge of the global topic matter in the chat.

For the actual annotation task, the annotators used Elsner's Java-based annotation client[3], which provides a graphical user interface that assists in the assignment of individual posts to conversation threads. The chat viewer interface is shown in Figure 3.2. Annotators are able to easily annotate new threads and associate posts with existing threads using a combination of keyboard shortcuts and dragging posts with the mouse. The entire chat session is shown in the left-hand pane. When a new post is annotated or when a previously-annotated post is selected, all posts marked as being in that thread are shown in the right-hand pane, thus providing an easy visual

[3]Available at `http://www.cs.brown.edu/~melsner/`

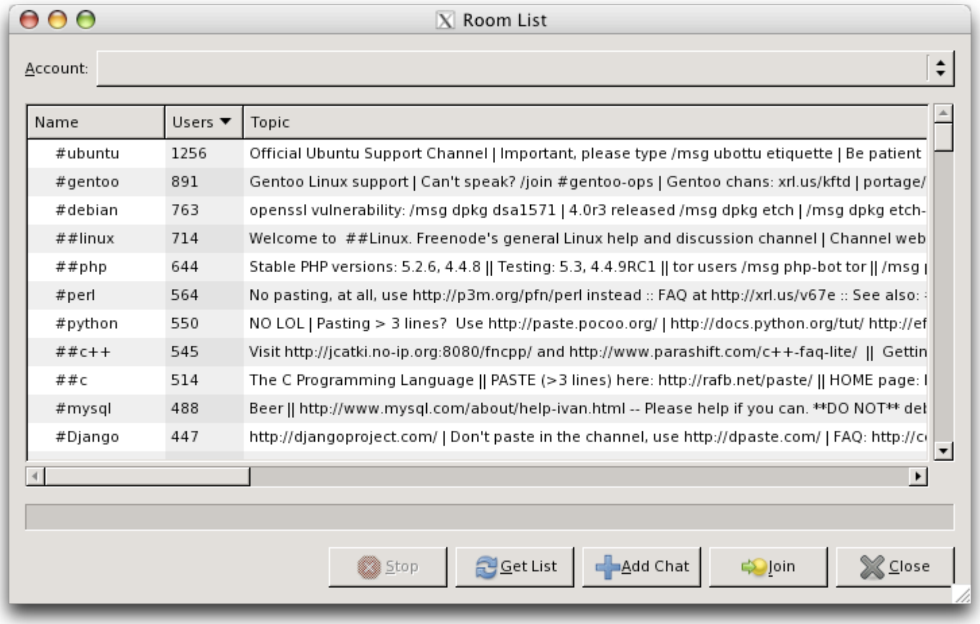


Figure 3.1: Freenode IRC server room list.

reference to the conversation.

The result of the annotation process is a text file comprising the text of each session, with each post prepended by an index corresponding to a conversation thread to which it belongs. These files are used directly in our maximum entropy classification process (see Section 3.4).

3.3 FIRST PHASE EXPERIMENTAL TECHNIQUES

As described in Wang *et al.* [25], we use a connectivity matrix to establish parent-child relationship between posts. Given our time-ordered sequence P of chat posts, where $P = \{p_i | time_{start} \leq i \leq time_{end}\}$ in a chat session, we construct a directed graph by creating an edge from p_j to all messages preceding it in time. The edge weights were derived from the cosine similarity of the word vectors of each post, which were constructed as described in Subsection 2.4.1. The initial graph is represented by the connectivity matrix W, where each element $w_{i,j}$ represents the weighted edge from p_i to p_j in the graph. The formal definition of the connectivity matrix is

$$w_{i,j} = \begin{cases} \frac{\vec{p}_i \cdot \vec{p}_j}{||\vec{p}_i||||\vec{p}_i||}, & \text{if } i > j \\ 0, & \text{otherwise} \end{cases}.$$

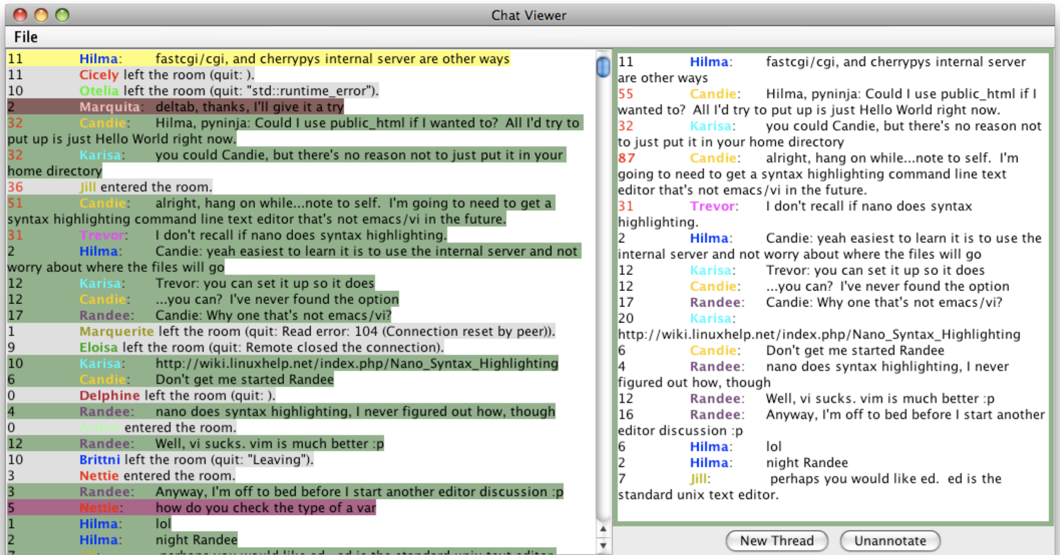


Figure 3.2: Chat viewer interface showing highlighted threads.

We use the initial connectivity matrix as a basis for finding links between pairs of messages. In the first stage, only cosine similarity between the TF-IDF weights is used for comparison. In latter stages, we augment the term vector and, in the case of considering distance between posts, we penalize the TF-IDF appropriately. This stages are described in detail in this section.

Many text processing tasks begin by employing stemming and/or stop word removal as a first step. Stemming involves removing the suffix of a word in order to consider only its root (*e.g., running* becomes *run*, *faded* becomes *fade*, etc.). Stop word removal involves discarding non-content bearing words such as function words (*e.g.,* conjunctions, prepositions, articles, etc.) or high-frequency words that occur too often in the text to provide useful distinguishing features (note that function words themselves are typically high frequency, so often techniques such as removal of the top 50 most frequent words will often do a good job at removing the function words). We have intentionally chosen not to employ stemming or stop word removal at this stage of our experiments. There are two primary reasons for this: 1) chat posts are sparse and often an entire post may consist of what might be considered non-content bearing words under other contexts, so we wish to preserve this in the hope that even the non-content bearing words, or specific morphologies of words might tend to assist in grouping like content; and 2) follow-on techniques such as WordNet hypernym augmentation (discussed in Subsection 3.3.2)

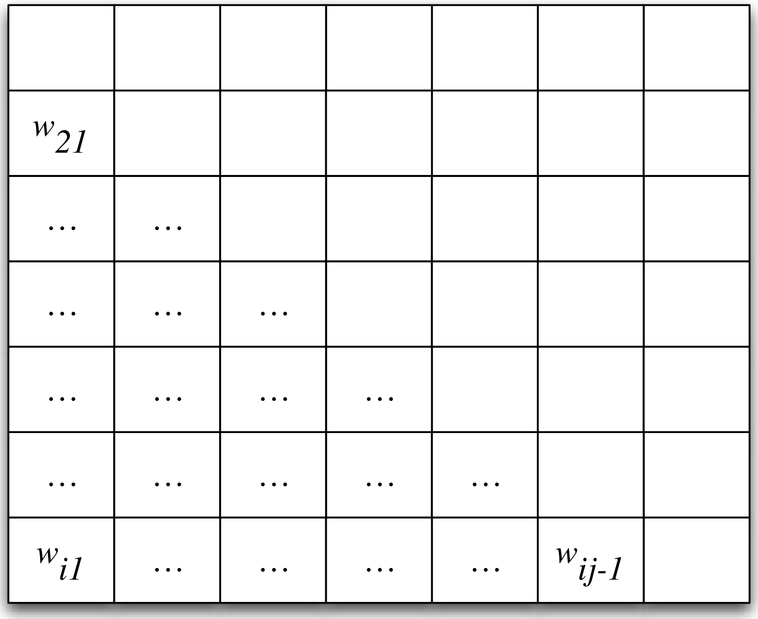


Figure 3.3: Illustration of connectivity matrix. Value w represents the weighted similarity of post i with post j.

provide automatic stemming of words, so it is not necessary to do so when building our initial connectivity matrix.

An additional decision that we made was to preserve punctuation and other non-word tokens to observe the effect that these items have on post similarity. We also included system messages, such 'PART' (displayed when a user departs the chat session) and 'JOIN' (displayed when a user joins the chat session) notifications, to observe the thread detection performance on these "known" related messages.

3.3.1 Time-Distance Penalization

For time-distance penalization, we consider that the further post j is from post i in a chat session (*i.e.,* the more posts that are interleaved between the two), the less likely the association between post i and post j in a particular topic thread.

We assign a simple penalization to our original weight as follows:

$$w'_{i,j} = w_{i,j} \left(\frac{1}{|i - j|} \right)$$

where $w'_{i,j}$ is our new time-distance penalized weighting factor for the edge between i and j.

3.3.2 Hypernym Augmentation

The intuition behind hypernym augmentation is that posts relating to the same subject may not include identical terms, though they may in fact include terms that are in the same semantic category. The Princeton WordNet[4] ontology includes hypernyms as one form of semantic relationship in its database.

A hypernym of a word is a word that is more generic than the given word. For example, 'canine' is more generic than 'dog,' thus 'canine' is a hypernym of 'dog.'

In our analysis, we consider each token in every post being evaluated. We augment the feature vector of the post with the next two levels of hypernyms of nouns and verbs found in the post. In deciding which hypernym path to follow, we chose the path from the first given sense as that is typically the most common usage of that word.

3.3.3 Nickname Augmentation

We are beginning to explore the relationship between the user and the topic thread. Our simplified initial model simply assigns the user nickname to the post feature vector. Thus, posts by the same user should be weighted more similarly to indicate the higher probability that they are part of the same conversation.

TD#	Description
1	TF-IDF only
2	TF-IDF + TDP
3	TF-IDF + HA
4	TF-IDF + TDP + HA
5	TF-IDF + HA + NA
6	TF-IDF + HA + NA + TDP

Table 3.3: Thread detection techniques. Key: TF-IDF - term frequency-inverse document frequency, HA - hypernym augmentation, NA - nickname augmentation, TDP - time-distance penalization.

The initial phase of our experiments used the original NPS Chat Corpus. It was divided into six groups, with each group implementing the feature sets shown in Table 3.3.

[4]Available from the Cognitive Science Laboratory at Princeton University: `http://wordnet.princeton.edu/`

3.3.4 Thread Extraction

The extraction of a conversation thread was accomplished by the algorithm shown in Figure 3.4. Given a root post, the algorithm returns all subsequent messages deemed to be a part of that the thread, as well as other threads that may spawn from the original thread. Future work will improve upon this algorithm; in particular, to recover threads that may have a broken link (*i.e.,* threads containing posts not having a similarity score above threshold) and to capture multiple parent conversations that may merge into a single thread.

$post\_queue =$ new queue
$post\_queue.add(root\_post)$
while $post\_queue$ not empty **do**
  get $post$ from $post\_queue$
  for each $< i, j >$ tuple from connectivity matrix **do**
    if $i = post$ and $weight_{ij} > threshold$ **then**
      $post\_queue.add(j)$
    end if
  end for
end while

Figure 3.4: Thread extraction algorithm

3.4 SECOND PHASE EXPERIMENTAL TECHNIQUES

In the second phase of our experiments, we examined the effects of maximum entropy classification on the IRC data collected from Freenode (##iphone, ##physics, and #python sessions). For comparison purposes, we elected to use the same methodology and statistics as in the Elsner and Charniak study [16,], although our feature construction approach differed slightly as shall be described.

This phase was conducted in two stages: one using the standard maximum entropy classifier and the second using the maximum entropy classifier augmented with LDA.

3.4.1 Maximum Entropy Classification

Elsner and Charniak's classification technique employs the MEGA Model Optimization Package maximum entropy classifier[5] written by Daumé. A full description of the classifier and its usage can be found on the website, along with a unpublished paper describing the algorithms

[5]Available from website of Hal Daumé III at the University of Utah School of Computing: `http://www.cs.utah.edu/~hal/megam/index.html`

employed.

The Elsner and Charniak experimental setup provides a Python “wrapper” around the Daumé classifier; several utility programs are used to construct the feature set and associated files. Training and testing are both done in a single step by passing the annotated training and testing chat sessions to the classifier as inputs (see Figure 3.5 for a graphical depiction of the classification process). Due to current limitations of the software, only one training file and one test file per classification cycle are permissable. To compensate for this, we used model averaging across the corpus using two different testing criteria, the details of which now follow.

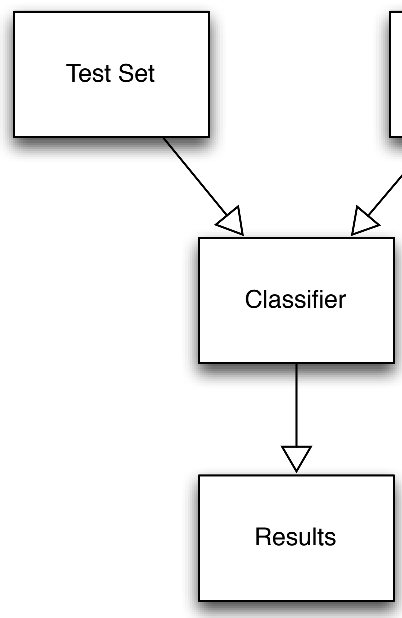


Figure 3.5: Maximum entropy classifier

We first trained the model on chat files from each annotator and tested against files annotated by *different* annotators for the same session; we then trained the model on chat files by each annotator from different sessions and tested against files annotated by the *same* annotator for *different* sessions. The primary objective in this two-pronged approach was to observe if a single-annotator training model performed comparably to human annotation by different annotators.

The actual steps taken in processing each file were as follows:

1. Unigram statistics were compiled for each file and the 50 most frequent words (stop

words) were removed.

2. A technical word list was compiled for each set of sessions by utilizing a source text related to the chat session topic matter and filtering out all words that were found in the Wall Street Journal texts of the Penn Treebank[6]. The intuition behind this step is that vocabulary related to the chosen chat session global topics would be unlikely to appear in the Wall Street Journal files as the articles predate (in the case of ##iphone and #python) or are more general (in the case of ##physics) than the technical material discussed in the chat session.

3. The model was constructed and evaluated by the maximum entropy classifier, once for each training-test pair.

4. Models were then evaluated using standard accuracy, precision, recall, and F-score values and were averaged across sessions for each of two testing criteria categories.

The following texts were used as source material for technical words for the sessions indicated:

Session	Text
##iphone	iPhone OS Programming Guide[7]
##physics	Newtonian Physics textbook[8]
#python	Dive into Python[9]

The technical word list tended to contain interesting results. For example, in the sample Linux technical word list included with the classifier, words such as *voip*, *chmod*, *inittab*, and *bashrc* were listed, but so too were words such as *thankyou* and *there's*, as well as different forms of numbers, symbols, and URLs. It is clear that proper tokenization (and perhaps error correction) plays a key role in the success of this method, as do appropriate choices for the technical and non-technical source texts.

[6] Details on the University of Penn. Dept. of Computer Science website at `http://www.cis.upenn.edu/~treebank/`

[7] Available from Apple, Inc., Developer Connection website at `http://developer.apple.com/iphone/library/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/iPhoneOSProgrammingGuide.pdf`

[8] Freely available textbook issued under the Creative Commons license. Available at `http://www.lightandmatter.com/area1book1.html`

[9] Freely available at `http://www.diveintopython.org/`

3.4.2 Maximum Entropy Classification with LDA Augmentation

The second stage of our maximum entropy experiments used an identical procedure to that described in the previous section, but with one important change: instead of compiling a technical words list as in step 2 in that procedure, we utilized LDA classification in an attempt to find vocabulary words groupings in latent topic areas in the chat (see Figure 3.6). As the technical words approach is a shallow attempt to describe a topic feature inherent in that chat, it is our hope that LDA will: 1) provide a more descriptive vocabulary based on the actual latent topics in the chat, and 2) eliminate the reliance on a technical document source.

For LDA classification, we used Steyvers and Griffiths's Topic Modeling Toolbox, version 1.3.2[10] under GNU Octave 3.0.0. Some preprocessing of the text was required to generate vocabulary and document indices. Utility modules that handle the required data formatting are provided as part of the Topic Modeling Toolbox and are trivial to use.

Steyver and Griffith's implementation of the LDA model is a variant of standard LDA as described in Chapter 2. In particular, this model places a symmetric Dirichlet prior, β,on the topic mixture. This parameter "smoothes [*sic*] the word distribution in every topic and can be interpreted as the prior observation count on the number of times words are samples from a document before any word from the topic is observed" [26].

As parameter estimation is an important factor in the success of the LDA algorithm, we fixed the number of topics T at 50 and used a known-good heuristic value of $50/T$ for the α parameter and varied β over $0.5^n, n = 1, \ldots, 10$. We then manually selected the topic grouping set which seemed to give the best description of the chat session. The groups that seemed to provide the best vocubulary groupings were those with β in the range of $0.5^2 \ldots 0.5^4$. Values greater than 0.5^6 resulted in fewer words returned due to the higher threshold for probability of occurrence.

[10]The Topic Modeling Toolbox, available at the Univ. of California Irvine Cognitive Sciences Department website: `http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm`, is designed for use with Matlab, but the LDA classification module is fully compatible with Octave.

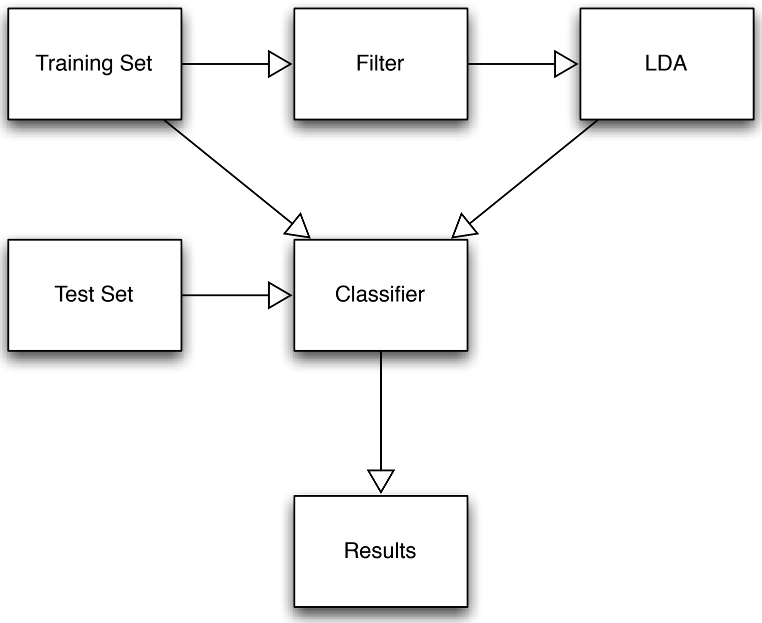


Figure 3.6: Maximum entropy classifier with LDA topic selection.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
RESULTS

In this chapter we present the results of our experiments as well as a discussion of their significance. We will begin first with observations regarding the chat corpus that we collected, along with insight gained from the annotation process. We will then discuss the results of the time-distance penalization experiments, followed by a a review of the performance of maximum-entropy classification and the effect of Latent Dirichlet Allocation on the classification process.

4.1 ANNOTATOR OBSERVATIONS

During the course of the annotation work, the annotators were encouraged to take notes and compile general observations regarding their findings. The following are some of those observations:

- Chat participants occasionally make standalone comments (typically with humorous intentions) that are either orthogonal to ongoing topics or during a lull in the conversation. This may motivate several turns of off-topic discussion or it may go unanswered. It some cases it appears that the motivation may be an attempt to end an “uncomfortable silence.”
- Posts that contain only emoticons tend to mark a single conversation.
- “Real” names are easier to keep track of during annotation than arbitrary user IDs.
- Attention words such as *hey* often mark the beginning of a schism or new conversation.
- Mentions were helpful in determining conversation threads, but some users tended to use them more than others. (Usage is user dependent.)
- Tacit knowledge of subject matter is often helpful in manual conversation disentanglement.
- Some questions only get partial answers or get no answer at all. In this case, the questioner will often repeat or rephrase the question. They will also use a follow up, such as *anyone?*.
- Multiple CIs may be an indicator of that a conversation is ending (the topic has “played out” and the participants are using CIs as “filler” material.)

- A conversation thread that has started to taper off may be revived by a new question or by a joke, both of which have the tendency of prolonging a conversation for several more turns.

- Chat room participants can often be divided into two categories: persistent cliques and transitory participants. Persistent cliques include participants who maintain a longer presence in a room and are usually involved in many conversation threads. Transitory participants tend to be more goal-oriented in their conversation and often join a chat room to ask a specific question, then leave upon receiving an answer. Persistent clique members are often characterized by being familiar with one another and having a more relaxed conversational style than transitory participants. As Figure 4.1 indicates, there is a correspondence between number of posts and number of conversations in which users are involved: those who post more are more likely to be involved in multiple conversations.

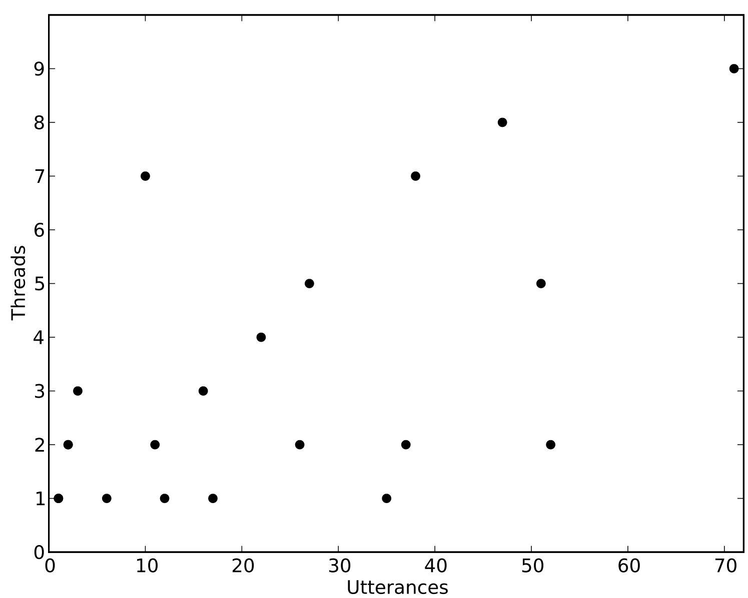


Figure 4.1: Utterances (posts) per speaker versus number of conversation threads in which speakers are engaged.

4.2 INTER-ANNOTATOR AGREEMENT

As discussed in Chapter 2, establishing inter-annotator agreement is an important factor in evaluating the performance of a classification model, since this establishes an upper bound on what can be expected from machine performance.

Summary results of the manual conversation thread annotation for the three chat rooms are show in Table 4.3 (see Appendix E for full annotation results).

Metric	Mean	Min	Max
1-to-1	**0.76301**	0.72527	0.81600
loc3	**0.90699**	0.88925	0.92403
M-to-1 (entropy)	**0.92232**	0.88502	0.95511
Avg. Conv. Length	16.83257	13.76255	19.35210
Avg. Conv. Density	1.23136	1.12836	1.34867
# Threads	28.56667	24.70000	33.20000
Entropy	3.53903	3.23833	3.90011

Table 4.1: Summary annotation metrics for ##iphone chat sessions.

Metric	Mean	Min	Max
1-to-1	**0.81652**	0.78891	0.85053
loc3	**0.93124**	0.91452	0.95219
M-to-1 (entropy)	**0.94263**	0.91145	0.97014
Avg. Conv. Length	24.32272	18.52808	31.25278
Avg. Conv. Density	1.13588	1.05863	1.22026
# Threads	14.76667	11.80000	17.70000
Entropy	2.50986	2.32122	2.67413

Table 4.2: Summary annotation metrics for ##physics chat sessions.

Metric	Mean	Min	Max
1-to-1	**0.74359**	0.69245	0.80493
loc3	**0.87330**	0.85220	0.89522
M-to-1 (entropy)	**0.87647**	0.84806	0.90293
Avg. Conv. Length	15.32323	13.76390	16.93643
Avg. Conv. Density	1.86632	1.73753	2.00879
# Threads	44.63333	40.40000	48.90000
Entropy	4.39527	4.19973	4.61509

Table 4.3: Summary annotation metrics for #python chat sessions.

As Elsner and Charniak [16] showed, and our inter-annotator agreement scores confirm, achieving consensus in conversation thread disentanglement can be a difficult task, even for human annotators. Each set of annotations by a particular annotator is a result of that individual's own theory of how the conversation mechanisms are being employed in that context. Even when involved in a conversation, human beings constantly use various cues – verbal and visual in the case of face-to-face, spoken conversation; textual and timing in the case of chat – that may not be evident in retrospect to a third party. Additionally, as described by Sacks *et al.* [10], humans regularly employ repair mechanisms during the course of a conversation to quickly repair

any turn-taking errors or misunderstandings. These facilities, of course, are not available to the annotator, though they may benefit from the chat participants' repair mechanisms should they recognize them as such.

For the annotation accomplished in this study, we can see that the lowest average score was for the #python chat session. We hypothesize that the reason for this is the highly technical nature of the discourse that, in many cases, required a higher level of tacit knowledge to follow the conversation. Code snippets and technical jargon were quite frequently shared between users; without having specific knowledge of the nature of the topics discussed, it presented a challenge to those trying to discern the flow of the conversation and to which thread each participant belonged. The ##physics sessions presented less of a challenge to our annotators as the conversations were generally more free-flowing and distinct. When new topics were introduced, they were often accompanied by enough context to allow the annotators to more easily follow the conversation.

4.3 TIME-DISTANCE PENALIZATION RESULTS

In this section, the evaluation approach used to study time distance penalization is presented first and a discussion of the results follows.

4.3.1 Evaluation

Standard precision, recall, and F-score measurement were used for evaluation of the results of the experiment. Results were hand-scored by examining the predicted message thread and marking each predicted post link as to whether or not it was an actual link (i.e., should have been included in the thread). A balanced F-score was used in these experiments. A weighted F-score might be preferred to weight precision over recall or vice versa, depending on actual application. As an example, a proposed application of topic detection would be to ensure compliance with security policy by sanitizing the session of topic threads that contain disallowed information. In this case, we would prefer to weight recall more highly, as it is more critical that we retrieve all the inappropriate conversation than it is that we be precise.

The measurements used for each are defined as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F\text{-}score = \frac{2(Precision \times Recall)}{Precision + Recall}$$

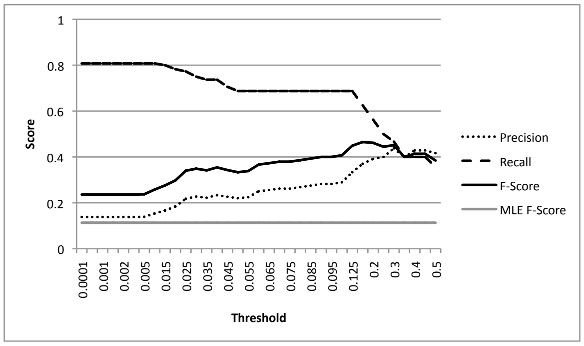
$TP =$ # of posts correctly scored as links within a thread
$FP =$ # of posts incorrectly scored as links within a thread
$FN =$ # of posts incorrectly not scored as links within a thread

4.3.2 Results

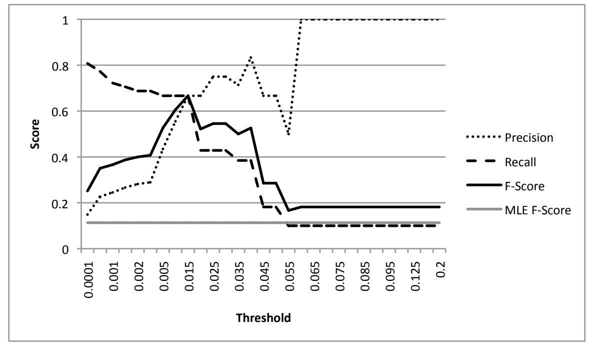
Figure 4.2 shows a comparison of our six thread detection schemes against a selected thread of interest. Note that the scale has been adjusted on the charts to better capture the threshold range. For the time-distance penalization charts, no posts were retrieved above a threshold of 0.2, so the chart is truncated at that value. A maximum likelihood estimate F-score was used as a baseline for comparison. The best performing detectors, with an F-score of 0.6667, were the ones that employed time-distance penalization together with TF-IDF, or with TF-IDF in combination with the other techniques. Against this particular thread, neither hypernym nor nickname augmentation made a significant difference in the detection results. Against two other threads tested we saw similar results, with F-scores for the TDP detectors consistently higher than those of the other detectors. More evaluation is needed across a more diverse data set to determine the consistency of this performance.

In Figure 4.3, we can see the effect that the time-distance penalization has on thread association. Subfigure 4.3(a) shows message posts with no time distance penalization. The two dense groupings are system messages—'PART' and 'JOIN' notifications—that do not belong to any chat conversation, thus they group only with themselves. In Subfigure 4.3(b), we observe that the time-distance penalization has the effect of "pulling apart" these strongly-linked messages since they occur further apart in the chat stream.

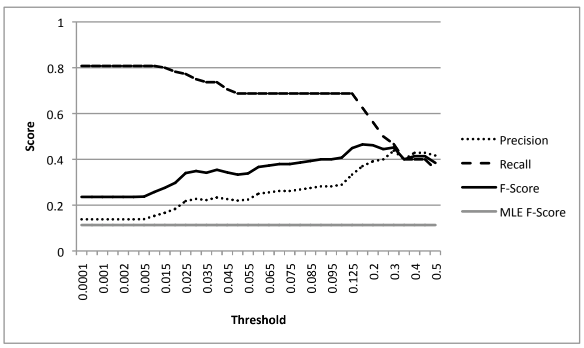
The effect on an actual conversation can be observed by noting the cluster in the upper right of Subfigure 4.3(a) surrounding post 104. This cluster represents "greeting" messages within the chat session (e.g. posts containing "hello,""hi," etc.). All such messages within the test block



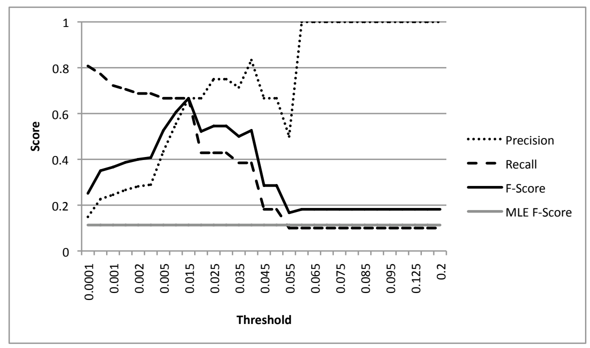
(a) TFIDF



(b) TFIDF+TDP

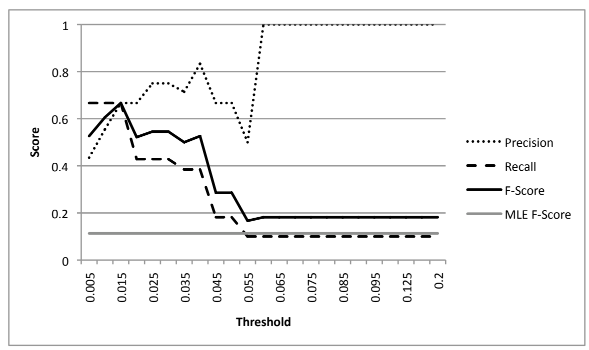


(c) TFIDF+HA



(d) TFIDF+HA+TDP

(e) TFIDF+HA+NA

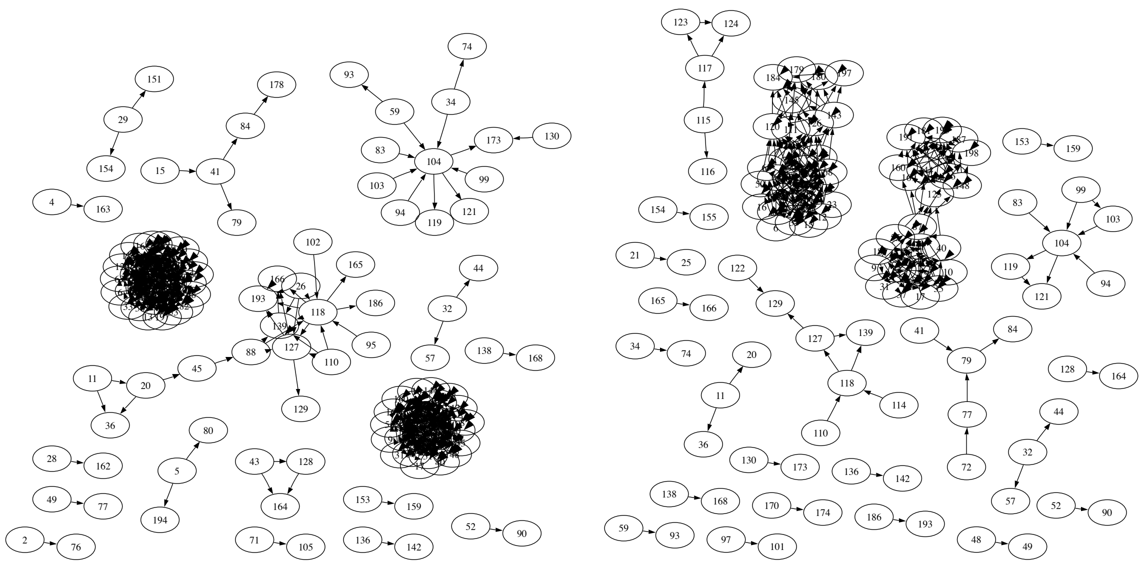


(f) TFIDF+HA+NA+TDP

Figure 4.2: Comparison of thread detection techniques across threshold values for selected thread of interest.

were linked in the same cluster, regardless of when they occurred within the session. In Subfigure 4.3(b), we can see that the cluster is smaller, with some links—34→74 and 130→173—removed from the initial grouping. This occurred due to those posts being part of a separate conversation, thus separated temporally from the others.

Our first-phase experiments quite clearly show the value of using time-distance as a feature in conversation thread extraction. In this set of experiments, combined with TF-IDF, it outper-



(a) No time-distance penalization (b) With time-distance penalization

Figure 4.3: Effect of time-distance penalization on chat post association.

formed other methods of thread classification, including hypernym augmentation and nickname augmentation.

An analysis of where the message thread prediction failed in these experiments shows several important results. One is the importance of tacit knowledge in a conversation. For example, one thread that we evaluated was a discussion of someone living in South Africa. When asked where the person lived, they responded "kwa zulu natal." Without tacit knowledge that KwaZulu Natal is a province of South Africa, it is not likely that this response would be automatically associated with the conversation thread based on the message content alone. There are several possible approaches to address this problem: 1) increase probability that the posts are associated because they occur within a certain timeframe, 2) increase probability that the posts are associated because they occur between two chat participants that we have already determined are involved in a conversation, or 3) augment our vocabulary with semantic information that includes, in this example, geographical data. In fact, an examination of WordNet 3.0 shows that *South Africa* and *KwaZulu-Natal* have a meronymy relationship: *South Africa HAS MEMBER KwaZulu-Natal*. This suggests that supplementing our feature vector meronymy information in addition to hypernymy information might yield better results. A problem with this approach is that the meronymy information in WordNet is sparse. As an example, the sense *car* is relatively well-populated with meronymy information and contains 29 *HAS PART* relationships, includ-

ing *air bag*, *gasoline engine*, *rear window*, etc., but it does not contain *steering wheel* or *clutch*, nor any of the other thousands of parts that comprise an automobile. The use of domain specific ontologies or automatic ontology building tools may help overcome this problem.

The example of *South Africa* also serves to highlight another shortfall in this approach. Our current method does not take collocations—word groupings—into account. Therefore, *South Africa* is seen as two separate tokens: *South* and *Africa*, so our algorithm would not search the *South Africa* taxonomy. There currently exists many excellent algorithms for collocation detection which may be a useful addition to our code.

The relatively simplistic method of increasing semantic content through hypernym augmentation yielded almost no gain in performance of our thread detector on any of the three threads tested. It is not evident that this methodology offers any advantages over other similarity scoring techniques such as Leacock-Chodorow or Resnik. Future experiments should employ one or more of these measures and evaluate the performance compared with hypernym augmentation.

The important detail learned from the first-phase experiments is that the time-distance penalization scheme, even in this relatively simple implementation, yields good results. Therefore, when building more advanced statistical models, the time-distance between posts is a factor that should not be overlooked in feature set construction.

4.4 MAXIMUM ENTROPY CLASSIFICATION RESULTS

In this section we provide overall and summary scores for the maximum entropy model and, for comparison, maximum entropy plus LDA scores (summary only). Full evaluation metrics for both models are provided in Appendix F and Appendix G. Final result accuracy is calculated using Elsner and Charniak's many-to-one entropy evaluation metric described in Chapter 2 and precision, recall, and F-score are as defined in the previous section.

4.4.1 Maximum Entropy Model Results

The results of using the maximum entropy classifier are shown in Tables 4.4, 4.5, 4.6, 4.7, 4.8, and 4.9. Summary results for all sessions are shown in Table 4.11. The average accuracy and F-score results were all in the same general range for all three chat topics, with the #physics chat sessions scoring slightly higher (in the 92 percent range). This correlates with the higher inter-annotator agreement scores that we saw for these files; we assess that this is due to the more conversational nature of the #physics chat with fewer technical "snippets," which made

the conversation threads easy to follow (thus leading to higher agreement).

Another item to note is, although the average accuracy and F-scores for the two different testing criteria (same annotator, different session and same session, different annotator), in all cases same session, different annotator scored slightly higher. This suggests a large degree in variety between feature sets session-to-session. Future work needs to be done to assess the relative session-to-session performance using different feature sets.

The most important finding of this experiment was that the maximum entropy classification scores approach those of human annotators, as shown in Table 4.10.

	Accuracy	Precision	Recall	F-score
Min	0.6819	0.6928	0.7804	0.7995
Max	0.9854	0.9875	1.0000	0.9927
Avg	0.8405	0.8618	0.9693	0.9093
Std Dev	0.0851	0.0864	0.0456	0.0522

Table 4.4: Classification results of same-annotator training and testing (different sessions) of ##iphone chat.

	Accuracy	Precision	Recall	F-score
Min	0.7019	0.7098	0.8715	0.8156
Max	0.9869	0.9869	1.0000	0.9934
Avg	0.8575	0.8707	0.9736	0.9178
Std Dev	0.0842	0.0792	0.0302	0.0530

Table 4.5: Classification results of same-session training and testing (different annotators) of ##iphone chat.

	Accuracy	Precision	Recall	F-score
Min	0.6409	0.6515	0.7452	0.7722
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.9202	0.9322	0.9852	0.9556
Std Dev	0.0881	0.0840	0.0370	0.0532

Table 4.6: Classification results of same-annotator training and testing (different sessions) of ##physics chat.

	Accuracy	Precision	Recall	F-score
Min	0.6600	0.6576	0.7451	0.7933
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.9259	0.9371	0.9833	0.9577
Std Dev	0.0864	0.0775	0.0481	0.0544

Table 4.7: Classification results of same-session training and testing (different annotators) of ##physics chat.

	Accuracy	Precision	Recall	F-score
Min	0.6109	0.5505	0.5064	0.6516
Max	0.8110	0.9433	0.9332	0.8278
Avg	0.7377	0.7794	0.7911	0.7780
Std Dev	0.0343	0.0742	0.0835	0.0331

Table 4.8: Classification results of same-annotator training and testing (different sessions) of #python chat.

	Accuracy	Precision	Recall	F-score
Min	0.7045	0.6764	0.6621	0.7266
Max	0.8101	0.8718	0.9316	0.8368
Avg	0.7583	0.7952	0.8011	0.7944
Std Dev	0.0297	0.0459	0.0717	0.0264

Table 4.9: Classification results of same-session training and testing (different annotators) of #python chat.

	Model Accuracy (Same Annot.)	Model Accuracy (Same Session)	Human Accuracy
##iphone	0.8405	0.8575	0.9223
##physics	0.9202	0.9259	0.9426
#python	0.7377	0.7583	0.8765

Table 4.10: Maximum entropy model versus human annotation accuracy.

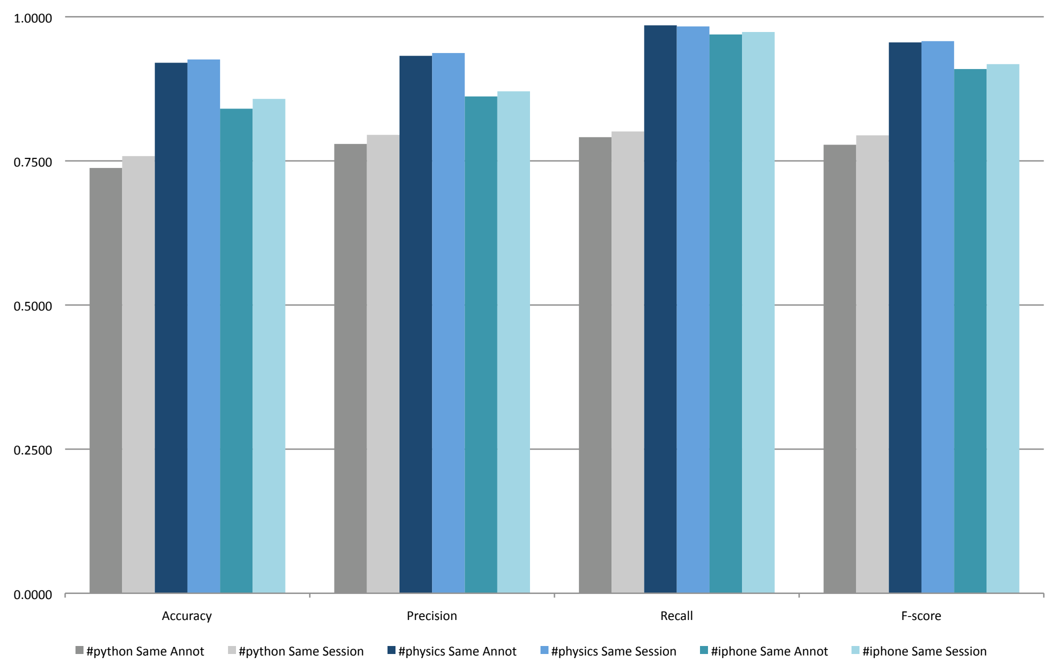


Figure 4.4: Comparison of classification results across all sessions using maximum entropy classifier.

Max-Ent Model

All Sessions, Same Annot Diff Day

	Accuracy	Precision	Recall	F-score
Min	0.6109	0.5505	0.5064	0.6516
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.8328	0.8578	0.9152	0.8810
Std Dev	0.0692	0.0815	0.0554	0.0461

All Sessions, All Same Day Diff Annot

	Accuracy	Precision	Recall	F-score
Min	0.6600	0.6576	0.6621	0.7266
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.8473	0.8677	0.9193	0.8900
Std Dev	0.0668	0.0675	0.0500	0.0446

Table 4.11: Classification results of same-session training and testing (different annotators) across all sessions using maximum entropy classification.

4.4.2 Maximum Entropy with LDA Augmentation Results

LDA augmentation of the maximum entropy classifier did not result in a significant difference in accuracy, precision, recall, or F-score metrics over the maximum entropy classifier alone. Note that scores across all sessions (as shown Figure 4.5) are virtually identical to the scores for the non-LDA classification (Figure 4.4). The explanation for this may be that the other features in the feature set outweigh the contribution of the technical words feature. More work should be done to assess the relative contribution of features to the model. Nonetheless, the fact that using LDA did not result in a significant decrease to the model's performance, combined with its lack of a requirement to provide technical and non-technical source texts, may still make it a promising alternative. Additionally, LDA was beneficial in its own right in order to illustrate and get a sense of the latent topics in the chat. LDA may be useful even in its own right for the auditing of chat files for sensitive material or for data mining purposes.

Max-Ent + LDA Model

All Sessions, Same Annot Diff Day

	Accuracy	Precision	Recall	F-score
Min	0.6076	0.5481	0.5231	0.6602
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.8314	0.8622	0.9135	0.8801
Std Dev	0.0696	0.0815	0.0589	0.0465

All Sessions, Same Day Diff Annot

	Accuracy	Precision	Recall	F-score
Min	0.6609	0.6582	0.6558	0.7267
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.8471	0.8675	0.9195	0.8900
Std Dev	0.0665	0.0673	0.0504	0.0444

Table 4.12: Classification results of same-session training and testing (different annotators) across all sessions using maximum entropy classification with LDA topic detection.

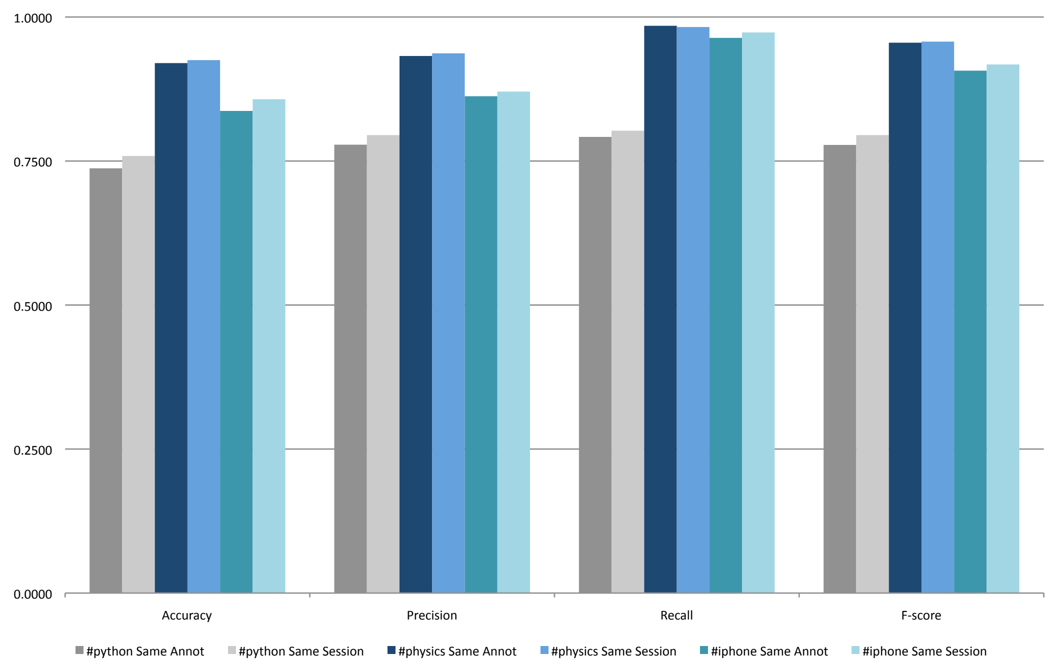


Figure 4.5: Comparison of classification results across all sessions using maximum entropy classifier with LDA.

4.4.3 Maximum Entropy Classification Summary

Maximum entropy classification proved to be an excellent technique for conversation thread classification, performing on par with human annotators. As our annotation has shown, conversation thread extraction is a difficult task even for human annotators, and the decision of whether a given pair of threads belong in the same conversation class is highly subjective. As in the Elsner and Charniak study, we have observed that annotators tend to be either "chunkers" or "splitters" – they have a predisposed proclivity toward grouping posts as conversations or separating them. Thus, it would be difficult to argue that much greater performance may be expected from maximum entropy classification, as it is already performing at the level of human annotators. Any further gain in improvement would likely be in tuning toward a single annotator's preferences, but this would be at the expense of a general model.

Perhaps because of the aforementioned maximum entropy performance, we did not see a notable change in accuracy by admitting LDA topic detection to our model. We do not believe that this invalidates the approach; rather, we believe that the relative performance of the model with and without LDA is more due to the higher performance of other features (*e.g.,* mentions and time-distance). Further studies should be conducted in order to confirm this theory and to quantify the relative contribution of these feature sets.

LDA does show remarkable promise in automatically extracting topic clusters. This could be useful in a broad range of applications, such as data mining or providing an automated auditing capability for chat logs. This method should be preferred to simple “clean/dirty word” lists, as it will capture a word within the context of a broader topic. Thus even words which appear benign in other contexts may become suspicious when appearing in a certain topic category.

An area where the use of LDA may be improved is in parameter estimation. As we have shown in this study, the use of previously determined α, β and θ parameters provides a good starting point for the application of the LDA algorithm. We elected to iterate over several β values that were likely to yield good results based on previous work in the field. A useful endeavor for future studies would involve techniques for better estimation of these LDA parameters. Additionally, the hLDA model should be investigated for possible use due to its ability to estimate the number of topics in the mixture without requiring a fixed parameter.

CHAPTER 5: CONCLUSIONS AND FUTURE WORK

In conclusion, our research shows that we can successfully perform automated topic detection and conversation extraction across many domains. Although we have achieved significant results, we are merely at the beginning stages of exploring the potential of statistical natural language processing techniques as it applies to chat and other forms of computer-mediated communications. As this work highlights, there are many possible applications, particularly in the realm of datamining and security.

To summarize our key findings, we showed the following:

- Conversation thread extraction is a difficult task for humans, as demonstrated by our inter-annotator agreement scores.
- The temporal distance between posts plays an important role in their classification. There is potential to improve the contribution of this feature by building more descriptive statistical models of the distribution of posts over time.
- Tacit knowledge is important to the discovery of semantic relationships which may influence the classification decision (implying a need for domain-specific ontologies).
- More research should be conducted into hypernym augmentation techniques using tools such as WordNet or other ontological databases.
- Maximum entropy is extremely effective technique for conversation thread classification.
- Latent Dirichlet Allocation can successfully find latent topics in chat, but more work needs to be done to fine tune the parameters to suit the domain.
- Although we collected many hours of chat data for this study, we believe that an even larger corpus with a wider variety of topics would be beneficial for further LDA study. To this end, we encourage further contributions of chat to this corpus.
- Continued exploration of feature set construction should be conducted. Natural language, hence chat, is a remarkably rich and diverse medium. We have only scratched the surface of its characteristics in this study. Future work should investigate incorporating work

such as Forsyth's part-of-speech and dialog-act tagging methodologies to enable more effective feature sets.

The aim of this work was to lay a solid foundation for future research into text classification of chat and the show the potential of advanced statistical techniques such as Latent Dirichlet Allocation and others to increase the value of text analysis tools to the warfighter. Our goals are to quickly get information to those who need it and present it in a manner that is useful, while denying it from those who do not. We believe this research moves us closer to achieving this end.

APPENDIX A: ACRONYMS AND ABBREVIATIONS

BPNN Back Propagation Neural Network

C2 Command and Control

C3 Command, Control, and Communication

C4I Command, Control, Communication, Computers, and Intelligence

CENTCOM United States Central Command

CI Chat Initialism

CRP Chinese Restaurant Process

CMC Computer-mediated Communication

DA Dialogue Act

GENSER General Service (related to communications)

HA Hypernym Augmentation

HMM Hidden Markov Model

HLDA Hierarchical Latent Dirichlet Allocation

IM Instant Messaging

IORNOC Indian Ocean Regional Network Operations Center

IRC	Internet Relay Chat
JTF	Joint Task Force
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
NA	Nickname Augmentation
NLP	Natural Language Processing
NORTHCOM	United States Northern Command
pLSI	Probabilistic Latent Semantic Indexing
POS	Part of Speech
PRNOC	Pacific Regional Network Operations Center
SI	Special Intelligence
SIT	Schism Inducing Turn
SVM	Support Vector Machines
TDP	Time-distance Penalization
XML	Extensible Markup Language

APPENDIX B:
GLOSSARY

The following is a list of potentially unfamiliar terms found in the text of the thesis. They are provided here for reference.

affiliation	the process by which a speaker or speakers attach themselves to a conversation
aside	comment that is produced to be marginal to the ongoing conversation; like toss-outs, they are topic-relevant and do not strongly implicate a response
bigram	a lexical unit comprising two words
chat room	a virtual domain comprising any number of chat participants, usually centered around a global topic or theme
chat initialism	abbreviations that are characteristic of computer-mediated communication, such as LOL (laughing out loud), BRB (be right back), etc.
disentanglement	the act of extracting conversation threads from a chat dialog
emoticon	a portmanteau of the words "emotion" and "icon"; a symbol, usually in ASCII text, that is meant to convey the emotional disposition of the writer; often used in reaction to another user's message
floor	a new conversation
global topic	the overall theme of a chat room (*e.g.,* Python programming, physics, etc.)

mention	the inclusion of a user's nickname in text to indicate to whom an utterance is directed
n**-gram**	a lexical unit comprising n words
nickname	user "handle" in a given chat session; many clients allow a user to maintain multiple nicknames and change these nicknames during the course of a chat session
persistent clique	a group of chat participants that are characterized by prolonged presence in a chat room and participating in a number of conversation threads over varying topics
post	an individual message from a user in a chat session; in this study posts comprise several data elements including user name, nickname (optional), time stamp, and message text
schism	the emergence of a new conversational thread amidst existing conversational threads
session	in the context of this study, a transcript from one particular chat room for a given time period
toss-out	an utterance that does not require response or acknowledgement, characterized by: 1) being topic relevant to the ongoing conversation, 2) being organizationally responsive to the in-progress conversation, 3) not targeting a specific recipient or recipients
transitory participant	a chat participant that enters a chat room for a short duration, usually in order to ask a question or get specific information
trigram	a lexical unit comprising three words

turn-taking	the process of determining which participant holds the floor in a conversation
unigram	a lexical unit comprising a single word
user	a participant in a chat session; may have one or more associated nicknames

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C: CHAT ABBREVIATIONS AND INITIALISMS

The following are the 50 most common chat acronyms and abbreviations with their meanings as listed by the NetLingo website[27].

2moro	Tomorrow
2nite	Tonight
BRB	Be Right Back
BTW	By The Way
B4N	Bye For Now
BCNU	Be Seeing You
BFF	Best Friends Forever
CYA	Cover Your Ass
DBEYR	Don't Believe Everything You Read
DILLIGAS	Do I Look Like I Give A Sh**
FUD	Fear, Uncertainty, and Disinformation
FWIW	For What It's Worth
GR8	Great
ILY	I Love You

IMHO	In My Humble Opinion
IRL	In Real Life
ISO	In Search Of
J/K	Just Kidding
L8R	Later
LMAO	Laughing My Ass Off
LOL	Laughing Out Loud -or- Lots Of Love
LYLAS	Love You Like A Sister
MHOTY	My Hat's Off To You
NIMBY	Not In My Back Yard
NP	No Problem
NUB	it stands for a new person
OIC	Oh, I See
OMG	Oh My God
OT	Off Topic
POV	Point Of View
RBTL	Read Between The Lines

ROTFLMAO	Rolling On The Floor Laughing My Ass Off
RT	Real Time
RTM	Read The Manual
RTFM	Read The F**king Manual
SH	Sh** Happens
SITD	Still In The Dark
SOL	Sh** Out of Luck
STBY	Sucks To Be You
STFU	Shut The F**k Up
SWAK	Sealed With A Kiss
TFH	Thread From Hell
THX	Thanks
TLC	Tender Loving Care
TMI	Too Much Information
TTYL	Talk To You Later
TYVM	Thank You Very Much
VBG	Very Big Grin

WEG	Wicked Evil Grin
WTF	What The F**k
WYWH	Wish You Were Here
XOXO	Hugs and Kisses

APPENDIX D: COMMONLY ENCOUNTERED CHAT EMOTICONS

The following is a list of emoticons commonly encountered in text-based chat. From: Netlingo [27].

<table>
<tr><td>@>--;--</td><td>A rose</td><td>%-6</td><td>All Mixed Up</td></tr>
<tr><td>O:-)</td><td>Angel</td><td>0*-)</td><td>Angel wink - female</td></tr>
<tr><td>0;-)</td><td>Angel wink - male</td><td>:-{</td><td>Angry</td></tr>
<tr><td>:-Z</td><td>Angry face</td><td>:-{{</td><td>Angry Very</td></tr>
<tr><td>>:-(</td><td>Annoyed</td><td>~:o</td><td>Baby</td></tr>
<tr><td>~~\8-O</td><td>Bad-Hair Day</td><td>d:-)</td><td>Baseball</td></tr>
<tr><td>:-)</td><td>Basic</td><td>:-{0</td><td>Basic Mustache</td></tr>
<tr><td>:~-(</td><td>Bawling</td><td>:-){</td><td>Beard</td></tr>
<tr><td>(:-{~</td><td>Beard - long</td><td>: =</td><td>Beaver</td></tr>
<tr><td>%-|</td><td>Been up All Night</td><td>:-)^<</td><td>Big Boy</td></tr>
<tr><td>(:-)</td><td>Big Face</td><td>:-)8<</td><td>Big Girl</td></tr>
<tr><td>(((H)))</td><td>Big Hug</td><td>:-X</td><td>Big Wet Kiss</td></tr>
<tr><td>=|:o}</td><td>Bill Clinton smiley</td><td>(:-D</td><td>Blabber Mouth</td></tr>
<tr><td>?-(</td><td>Black Eye</td><td>(:-</td><td>Blank Expression</td></tr>
<tr><td>#-)</td><td>Blinking</td><td>:-]</td><td>Blockhead</td></tr>
<tr><td>:-!</td><td>Bored</td><td>I:(</td><td>Botox smiley</td></tr>
<tr><td>:-}X</td><td>Bow Tie-Wearing</td><td><|:-)>=</td><td>Boy Scout</td></tr>
<tr><td>%-6</td><td>Brain Dead</td><td>:-(=)</td><td>Bucktoothed</td></tr>
<tr><td>:-E</td><td>Bucktoothed Vampire</td><td>:-F</td><td>Bucktoothed Vampire with One Tooth Missing</td></tr>
<tr><td>:-#|</td><td>Bushy Mustache</td><td>}|{</td><td>Butterfly</td></tr>
<tr><td>})i({</td><td>Butterfly (prettier)</td><td>}:-X</td><td>Cat</td></tr>
<tr><td>q:-)</td><td>Catcher</td><td>C=:-)</td><td>Chef</td></tr>
<tr><td>8^</td><td>Chicken</td><td>;-(</td><td>Chin up</td></tr>
<tr><td>*<<<<+</td><td>Christmas Tree</td><td>:-.)</td><td>Cindy Crawford</td></tr>
</table>

<table>
<tr><td>*<):o)</td><td>Clown</td><td>:-8(</td><td>Condescending Stare</td></tr>
<tr><td>:-S</td><td>Confused</td><td>%)</td><td>Confused</td></tr>
<tr><td>:-{</td><td>Count Dracula</td><td>H-)</td><td>Cross-Eyed</td></tr>
<tr><td>:`-(</td><td>Crying</td><td>:*(</td><td>Crying softly</td></tr>
<tr><td>&:-)</td><td>Curly Hair</td><td>:-@!</td><td>Cursing</td></tr>
<tr><td>O-)</td><td>Cyclops</td><td>>:-></td><td>Devilish</td></tr>
<tr><td>:-e</td><td>Disappointed</td><td>%-}</td><td>Dizzy</td></tr>
<tr><td>:3-]</td><td>Dog</td><td>:-))</td><td>Double Chin</td></tr>
<tr><td>:*)</td><td>Drinking every night</td><td>:-B</td><td>Drooling out of Both Sides of Mouth</td></tr>
<tr><td>.\/</td><td>Duck</td><td><:-l</td><td>Dunce</td></tr>
<tr><td>:-6</td><td>Eating Something Spicy</td><td>(:-|</td><td>Egghead</td></tr>
<tr><td>5:-)</td><td>Elvis</td><td>:")</td><td>Embarrased</td></tr>
<tr><td>:-}</td><td>Embarrassed Smile</td><td>0|-)</td><td>Enjoying the Sun</td></tr>
<tr><td>>:)</td><td>Evil</td><td>>-)</td><td>Evil Grin</td></tr>
<tr><td>G(-'.'G)</td><td>Fighting Kid</td><td>l:-O</td><td>FlatTop Loudmouth</td></tr>
<tr><td>=:-H</td><td>Football player</td><td>:-W</td><td>Forked Tongue</td></tr>
<tr><td>:^{=</td><td>Frank Zappa</td><td>%*@:-)</td><td>Freaking Out</td></tr>
<tr><td>/:-)</td><td>Frenchman with a beret</td><td>8)</td><td>Frog</td></tr>
<tr><td>:-<</td><td>Frowning</td><td>):-(</td><td>Frowning Smiley with Hair</td></tr>
<tr><td>:-/</td><td>Frustrated</td><td>=:-)</td><td>Funny Hair</td></tr>
<tr><td>*:*</td><td>Fuzzy</td><td>*:*}</td><td>Fuzzy With a Mustache</td></tr>
<tr><td>~~:-(</td><td>Getting Rained On</td><td>8*)</td><td>Glasses and a Half Mustache</td></tr>
<tr><td>{:-)</td><td>Hair Parted in the Middle</td><td>}:-)</td><td>Hair Parted in the Middle Sticking up on Sides</td></tr>
<tr><td>:-})</td><td>Handlebar Mustache</td><td>%-)</td><td>Happy Drunk</td></tr>
<tr><td>:-'</td><td>Has a Dimple</td><td>:%)%</td><td>Has Acne</td></tr>
<tr><td>:-#</td><td>Has Braces</td><td>:(#)</td><td>Has Braces variation</td></tr>
<tr><td>:-`|</td><td>Have a Cold</td><td>|:-)</td><td>Heavy Eyebrows</td></tr>
<tr><td>/;-)</td><td>Heavy Eyebrows - Slanted</td><td>l^o</td><td>Hepcat</td></tr>
<tr><td>(_8(|)</td><td>Homer Simpson</td><td>(_8^(|)</td><td>Homer Simpson</td></tr>
<tr><td>(^~^~^)</td><td>Hot Ass Walking Away</td><td>*^_^*</td><td>Huge Dazzling Grin</td></tr>
<tr><td>:0</td><td>Hungry</td><td>o[-<]:</td><td>I am a skater or I like to skate</td></tr>
<tr><td>%*}</td><td>Inebriated</td><td>(|):-)=II=</td><td>Jewish Blonde</td></tr>
</table>

?:^[]	Jim Carrey	(8 {	John Lennon
@:-}	Just Back From Hairdresser	:-T	Keeping a Straight Face
:-x	Kiss	:-*	Kiss on the cheek
>^,,^<	Kitty Cat	&	Kitty cleaning a hind paw
~*=	Kitty running away from you	:p	Kitty with tongue hanging out
@(*0*)@	Koala	:-D	Laughing
%OD	Laughing like crazy	(-:	Left Hand
?-:	Lefthanded tongue touching nose	>;->	Lewd Remark
:-9	Licking Lips	;-,	Like, Duh
:-x	Lips are Sealed	8:-)	Little Girl
%-)	Long Bangs	%+{	Lost a Fight
|-(	Lost Contact Lenses	X-(	Mad
;-(	Mad Look	&-l	Makes Me Cry
:-(*)	Makes Me Sick	:-S	Makes No Sense
@@@@:-)	Marge Simpson	@|-)	Meditating Smiley
#:-)	Messy Hair	8(:-)	Mickey Mouse
:)	Midget	~~:-(	Mohawk
:-{	Mustache	:-{)=	Mustache & Goatee
:-3	Mustache (Handlebar Type)	{:-{)}	Mustache and Beard
:-#	My Lips Are Sealed	(-)	Needs Haircut
):-(	Nordic	:/)	Not Amused
8-O	Omigod	:=)	Orangutan
:-?	Pensive	:^)	Personality
3:]	Pet Dog	:8)	Pig
:---)	Pinnochio	P-(	Pirate
3:[	Pitbull	:-<	Pointy Mustache
}:^#)	Pointy Nosed	+<:-)	Pope
:-t	Pouting	:-[	Pouting variation
+:-)	Priest	;~[	Prizefighter
X:-)	Propeller Head	?-)	Proud of black eye
=:-)	Punk	=:-(	Punk Not Smiling
<(-'.'-)>	Puppy dog	:-r	Rasberry

<table>
<tr><td>:-C</td><td>Real Unhappy</td><td>~:-(</td><td>Really Bummed Out</td></tr>
<tr><td>:-))</td><td>Really Happy</td><td>([(</td><td>Robocop</td></tr>
<tr><td>[:]</td><td>Robot</td><td>@};---</td><td>Rose</td></tr>
<tr><td>3:*></td><td>Rudolph the red nose reindeer</td><td>:-(</td><td>Sad</td></tr>
<tr><td>:(</td><td>Sad Turtle</td><td>:-d</td><td>Said with a smile</td></tr>
<tr><td>:-y</td><td>Said with a Smile variation</td><td>M:-)</td><td>Saluting</td></tr>
<tr><td>*<|:-)</td><td>Santa Claus</td><td>:-></td><td>Sarcastic</td></tr>
<tr><td>:-@</td><td>Screaming</td><td>8-)</td><td>Scuba Diver</td></tr>
<tr><td>)8-)</td><td>Scuba Diver with Hair</td><td>$__$</td><td>Sees Money</td></tr>
<tr><td>:-i</td><td>Semi-Smile</td><td>,:-)</td><td>Shaved Left Eyebrow</td></tr>
<tr><td>8-O</td><td>Shocked</td><td>+-(</td><td>Shot Between the Eyes</td></tr>
<tr><td>:-V</td><td>Shouting</td><td>:O</td><td>Singing</td></tr>
<tr><td>~:-P</td><td>Single Hair</td><td>:-/</td><td>Skeptical</td></tr>
<tr><td>':-/</td><td>Skeptical again</td><td>:-7</td><td>Skeptical variation</td></tr>
<tr><td>O-)</td><td>Smiley After Smoking a Banana</td><td>):-)</td><td>Smiley with Hair</td></tr>
<tr><td>:-,</td><td>Smirk</td><td>;^)</td><td>Smirking</td></tr>
<tr><td>:-i</td><td>Smoking a cig</td><td>:-?</td><td>Smoking a pipe</td></tr>
<tr><td>:-Q</td><td>Smoking while talking</td><td>~~~~8}</td><td>Snake</td></tr>
<tr><td>:-(<|</td><td>Standing Firm</td><td>=%-O</td><td>Stared at Computer Way Too Long</td></tr>
<tr><td>%-)</td><td>Staring at a Screen for 15 hours</td><td>(8-{)}</td><td>Sunglasses, Mustache, Beard</td></tr>
<tr><td>:0</td><td>Surprised</td><td>`:-)</td><td>Sweating</td></tr>
<tr><td>,:-)</td><td>Sweating on the Other Side</td><td>:-0</td><td>Talkative</td></tr>
<tr><td>:-S</td><td>Talking Gibberish</td><td>&-|</td><td>Tearful</td></tr>
<tr><td>-(:)(0)=8</td><td>Teletubby</td><td>:-)---</td><td>Thin as a Pin</td></tr>
<tr><td>%-\</td><td>Tired</td><td>:-?</td><td>Tongue Sticking Out</td></tr>
<tr><td>:-&</td><td>Tongue Tied</td><td>:-a</td><td>Tongue Touching Nose</td></tr>
<tr><td>*!#*!^*&:-</td><td>Total Head Case</td><td>}(:-(</td><td>Toupee Blowing in Wind</td></tr>
<tr><td>:-)))</td><td>Triple Chin</td><td><:>==</td><td>Turkey</td></tr>
<tr><td>x:-/</td><td>Uncertain</td><td>=):-)</td><td>Uncle Sam</td></tr>
<tr><td>:-\</td><td>Undecided</td><td>:-|</td><td>Unfazed</td></tr>
<tr><td>|:-)</td><td>Unibrow</td><td>|:-|</td><td>Unyielding</td></tr>
</table>

:-[	Vampire	:-)))	Very Happy
%')	Very Tired	(:-(	Very Unhappy
:-<	Walrus	@:-)	Wavy Hair
{(:-)	Wearing a Toupee	[:-)	Wearing a Walkman
8-)	Wearing Contacts	B-)	Wearing Glasses
:-{}	Wearing Lipstick	]-I	Wearing Sunglasses
{:-)	Wears a Toupee	:-1	Whatever
:-"	Whistling	;^?	Wigged Out
'-)	Winking	,-)	Winking Happy
;-)	Winking variation	#-)	Wiped out, partied all night
8<:-)	Wizard	-=#:-) \	Wizard with Wand
:-)8 :	Woman	:~)	Wondering
,-}	Wry and Winking	:-7	Wry Face
l-O	Yawning	|^o	Yawning or Snoring variation
:-(0)	Yelling	=8-0	Yikes

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E:
INTER-ANNOTATOR AGREEMENT RESULTS

The following are the full inter-annotator agreement results (three annotators) for three different chat rooms: ##iphone, ##physics, and #python. Sessions were logged July 17-31, 2008.

E.1 ##iphone METRICS

Session	Metric	Mean	Min	Max
17-Jul	1-to-1	0.84185	0.78832	0.91971
	loc3	0.92040	0.91045	0.93532
	M-to-1 (entropy)	0.98054	0.95620	1.00000
	Avg. Conv. Length	9.20024	6.52381	10.53846
	Avg. Conv. Density	1.18735	1.02920	1.48905
	# Threads	15.66667	13.00000	21.00000
	Entropy	2.87433	2.63738	3.28785
18-Jul	1-to-1	0.65185	0.63419	0.68205
	loc3	0.88278	0.84765	0.90435
	M-to-1 (entropy)	0.85698	0.78291	0.91111
	Avg. Conv. Length	38.21429	27.85714	45.00000
	Avg. Conv. Density	1.32764	1.10427	1.47692
	# Threads	16.00000	13.00000	21.00000
	Entropy	2.43688	2.10202	3.00043
19-Jul	1-to-1	0.75936	0.69920	0.86096
	loc3	0.93408	0.92438	0.94676
	M-to-1 (entropy)	0.95766	0.93182	0.97594
	Avg. Conv. Length	28.14725	22.00000	32.52174
	Avg. Conv. Density	1.20900	1.06551	1.45856
	# Threads	27.33333	23.00000	34.00000
	Entropy	3.09611	2.75017	3.78377
21-Jul	1-to-1	0.74676	0.72081	0.79865
	loc3	0.92819	0.92177	0.93311
	M-to-1 (entropy)	0.94360	0.89002	0.98646

Continued...

Session	Metric	Mean	Min	Max
	Avg. Conv. Length	15.15713	13.43182	16.88571
	Avg. Conv. Density	1.22617	1.13367	1.28088
	# Threads	39.33333	35.00000	44.00000
	Entropy	3.91289	3.33856	4.31152
22-Jul	1-to-1	0.69586	0.69112	0.70414
	loc3	0.87332	0.86540	0.88638
	M-to-1 (entropy)	0.87377	0.84970	0.89467
	Avg. Conv. Length	20.76914	16.25000	26.40625
	Avg. Conv. Density	1.31240	1.24763	1.40284
	# Threads	42.33333	32.00000	52.00000
	Entropy	4.47508	4.11199	4.81126
23-Jul	1-to-1	0.82711	0.78838	0.90041
	loc3	0.88702	0.86415	0.92157
	M-to-1 (entropy)	0.92531	0.90041	0.95436
	Avg. Conv. Length	7.74313	6.88571	8.31034
	Avg. Conv. Density	1.22545	1.12033	1.33195
	# Threads	31.33333	29.00000	35.00000
	Entropy	4.10224	3.96632	4.27904
24-Jul	1-to-1	0.59928	0.52587	0.67870
	loc3	0.84326	0.80837	0.87681
	M-to-1 (entropy)	0.82671	0.78580	0.86522
	Avg. Conv. Length	16.55520	14.57895	18.46667
	Avg. Conv. Density	1.35018	1.26233	1.40433
	# Threads	50.66667	45.00000	57.00000
	Entropy	4.42245	3.99141	4.80779
28-Jul	1-to-1	0.84986	0.82231	0.89669
	loc3	0.95723	0.94142	0.96932
	M-to-1 (entropy)	0.93939	0.89256	0.98760
	Avg. Conv. Length	12.83198	11.52381	14.23529
	Avg. Conv. Density	1.08953	1.05785	1.12810
	# Threads	19.00000	17.00000	21.00000
	Entropy	3.20827	3.02162	3.33637

Continued...

Session	Metric	Mean	Min	Max
29-Jul	1-to-1	0.80967	0.77341	0.84592
	loc3	0.96612	0.95528	0.97256
	M-to-1 (entropy)	0.97382	0.95166	0.99396
	Avg. Conv. Length	14.40948	13.79167	15.04545
	Avg. Conv. Density	1.31621	1.26284	1.35045
	# Threads	23.00000	22.00000	24.00000
	Entropy	3.11823	2.87153	3.47883
31-Jul	1-to-1	0.84848	0.80909	0.87273
	loc3	0.87747	0.85358	0.89408
	M-to-1 (entropy)	0.94545	0.90909	0.98182
	Avg. Conv. Length	5.29791	4.78261	6.11111
	Avg. Conv. Density	1.06970	1.00000	1.16364
	# Threads	21.00000	18.00000	23.00000
	Entropy	3.74380	3.59235	3.90423
Avg. All Sessions	1-to-1	0.76301	0.72527	0.81600
	loc3	0.90699	0.88925	0.92403
	M-to-1 (entropy)	0.92232	0.88502	0.95511
	Avg. Conv. Length	16.83257	13.76255	19.35210
	Avg. Conv. Density	1.23136	1.12836	1.34867
	# Threads	28.56667	24.70000	33.20000
	Entropy	3.53903	3.23833	3.90011

E.2 ##physics METRICS

Session	Metric	Mean	Min	Max
17-Jul	1-to-1	0.96020	0.94030	0.98507
	loc3	0.96181	0.94271	0.97917
	M-to-1 (entropy)	1.00000	1.00000	1.00000

Continued...

Session	Metric	Mean	Min	Max
	Avg. Conv. Length	23.07778	13.40000	33.50000
	Avg. Conv. Density	1.00498	1.00000	1.01493
	# Threads	3.33333	2.00000	5.00000
	Entropy	0.28726	0.11191	0.52639
18-Jul	1-to-1	0.95286	0.92929	0.98990
	loc3	0.96528	0.94792	1.00000
	M-to-1 (entropy)	1.00000	1.00000	1.00000
	Avg. Conv. Length	12.35000	8.25000	19.80000
	Avg. Conv. Density	1.00000	1.00000	1.00000
	# Threads	9.33333	5.00000	12.00000
	Entropy	2.06927	1.91439	2.15681
19-Jul	1-to-1	0.87367	0.85160	0.89269
	loc3	0.92797	0.92184	0.93487
	M-to-1 (entropy)	0.95053	0.93151	0.96804
	Avg. Conv. Length	17.50206	14.12903	20.85714
	Avg. Conv. Density	1.09665	1.07078	1.13014
	# Threads	25.66667	21.00000	31.00000
	Entropy	3.28143	3.16197	3.38866
21-Jul	1-to-1	0.72840	0.69801	0.77778
	loc3	0.92402	0.90987	0.94611
	M-to-1 (entropy)	0.95062	0.90741	0.97436
	Avg. Conv. Length	35.75000	29.25000	39.00000
	Avg. Conv. Density	1.39364	1.07407	1.83048
	# Threads	20.00000	18.00000	24.00000
	Entropy	3.06059	2.77389	3.55838
22-Jul	1-to-1	0.97044	0.95567	0.98030
	loc3	0.98667	0.98000	0.99000
	M-to-1 (entropy)	1.00000	1.00000	1.00000
	Avg. Conv. Length	16.99553	15.61538	18.45455
	Avg. Conv. Density	1.00493	1.00493	1.00493
	# Threads	12.00000	11.00000	13.00000
	Entropy	2.74950	2.66137	2.83381

Continued. . .

Session	Metric	Mean	Min	Max
23-Jul	1-to-1	0.98783	0.98540	0.99270
	loc3	0.99005	0.98507	0.99751
	M-to-1 (entropy)	1.00000	1.00000	1.00000
	Avg. Conv. Length	11.46989	10.53846	12.45455
	Avg. Conv. Density	1.04380	1.04380	1.04380
	# Threads	12.00000	11.00000	13.00000
	Entropy	2.34740	2.30592	2.37544
24-Jul	1-to-1	0.61024	0.58748	0.65434
	loc3	0.87937	0.85143	0.90476
	M-to-1 (entropy)	0.85301	0.78805	0.91607
	Avg. Conv. Length	33.81389	29.29167	37.00000
	Avg. Conv. Density	1.21764	1.19203	1.25462
	# Threads	21.00000	19.00000	24.00000
	Entropy	3.09977	2.50741	3.51298
28-Jul	1-to-1	0.58522	0.54209	0.64682
	loc3	0.80349	0.76171	0.85744
	M-to-1 (entropy)	0.82067	0.77207	0.89528
	Avg. Conv. Length	25.87831	18.03704	37.46154
	Avg. Conv. Density	1.34634	1.09240	1.53799
	# Threads	20.66667	13.00000	27.00000
	Entropy	3.32424	3.76419	2.83240
29-Jul	1-to-1	0.61574	0.55754	0.66071
	loc3	0.93835	0.93014	0.95476
	M-to-1 (entropy)	0.90146	0.81548	0.96429
	Avg. Conv. Length	57.72308	38.76923	84.00000
	Avg. Conv. Density	1.10913	1.00000	1.20238
	# Threads	9.66667	6.00000	13.00000
	Entropy	2.27516	1.55764	2.87409
31-Jul	1-to-1	0.88056	0.84167	0.92500
	loc3	0.93542	0.91453	0.95726
	M-to-1 (entropy)	0.95000	0.90000	0.98333
	Avg. Conv. Length	8.66667	8.00000	10.00000

Continued. . .

Session	Metric	Mean	Min	Max
	Avg. Conv. Density	1.14167	1.10833	1.18333
	# Threads	14.00000	12.00000	15.00000
	Entropy	2.60400	2.45350	2.68235
Avg. All Sessions	1-to-1	0.81652	0.78891	0.85053
	loc3	0.93124	0.91452	0.95219
	M-to-1 (entropy)	0.94263	0.91145	0.97014
	Avg. Conv. Length	24.32272	18.52808	31.25278
	Avg. Conv. Density	1.13588	1.05863	1.22026
	# Threads	14.76667	11.80000	17.70000
	Entropy	2.50986	2.32122	2.67413

E.3 #python METRICS

Session	Metric	Mean	Min	Max
17-Jul	1-to-1	0.63364	0.52632	0.76471
	loc3	0.86042	0.80937	0.89271
	M-to-1 (entropy)	0.79360	0.70588	0.85449
	Avg. Conv. Length	13.93671	10.76667	17.00000
	Avg. Conv. Density	1.78844	1.45201	2.20124
	# Threads	24.00000	19.00000	30.00000
	Entropy	3.56414	3.51237	3.62109
18-Jul	1-to-1	0.67831	0.62709	0.76257
	loc3	0.89995	0.86723	0.94109
	M-to-1 (entropy)	0.87058	0.83939	0.90084
	Avg. Conv. Length	18.01672	15.23404	20.45714
	Avg. Conv. Density	2.03911	1.85335	1.95158
	# Threads	40.33333	35.00000	47.00000

Continued. . .

Session	Metric	Mean	Min	Max
	Entropy	4.24396	4.00140	4.68757
19-Jul	1-to-1	0.69656	0.62772	0.79891
	loc3	0.86054	0.85766	0.86221
	M-to-1 (entropy)	0.85371	0.82473	0.90082
	Avg. Conv. Length	14.30940	13.38182	15.65957
	Avg. Conv. Density	1.89402	1.59103	2.34918
	# Threads	51.66667	47.00000	55.00000
	Entropy	4.52339	4.24990	4.75857
21-Jul	1-to-1	0.73560	0.69688	0.78470
	loc3	0.83657	0.81697	0.85633
	M-to-1 (entropy)	0.87866	0.81303	0.93201
	Avg. Conv. Length	18.27788	17.65000	19.08108
	Avg. Conv. Density	1.67705	1.96034	1.44193
	# Threads	38.66667	37.00000	40.00000
	Entropy	4.25823	4.14085	4.47135
22-Jul	1-to-1	0.76693	0.71615	0.81250
	loc3	0.88874	0.86318	0.90240
	M-to-1 (entropy)	0.90148	0.89583	0.90885
	Avg. Conv. Length	16.89167	15.36000	17.86047
	Avg. Conv. Density	2.21267	2.16276	2.27865
	# Threads	45.66667	43.00000	50.00000
	Entropy	4.35211	4.13655	4.64556
23-Jul	1-to-1	0.79864	0.76599	0.84218
	loc3	0.89921	0.88342	0.92441
	M-to-1 (entropy)	0.87438	0.84490	0.89388
	Avg. Conv. Length	18.30037	15.63830	20.41667
	Avg. Conv. Density	1.86939	1.70068	2.11020
	# Threads	40.66667	36.00000	47.00000
	Entropy	4.34544	4.26055	4.45993
24-Jul	1-to-1	0.74492	0.72065	0.76226
	loc3	0.85605	0.84975	0.86418
	M-to-1 (entropy)	0.84596	0.83655	0.85290

Continued. . .

Session	Metric	Mean	Min	Max
	Avg. Conv. Length	10.42071	9.89706	10.68254
	Avg. Conv. Density	2.06389	1.91679	2.34621
	# Threads	64.66667	63.00000	68.00000
	Entropy	4.99559	4.83134	5.09911
28-Jul	1-to-1	0.75805	0.71733	0.81676
	loc3	0.85640	0.82882	0.90823
	M-to-1 (entropy)	0.91288	0.89773	0.93182
	Avg. Conv. Length	14.18003	12.13793	18.05128
	Avg. Conv. Density	1.68371	1.50142	1.79688
	# Threads	51.33333	39.00000	58.00000
	Entropy	4.77518	4.34111	5.03267
29-Jul	1-to-1	0.78671	0.72697	0.81742
	loc3	0.88814	0.87205	0.90067
	M-to-1 (entropy)	0.89838	0.89280	0.90787
	Avg. Conv. Length	15.18013	14.92500	15.30769
	Avg. Conv. Density	1.78727	1.71859	1.83752
	# Threads	39.33333	39.00000	40.00000
	Entropy	3.98693	3.75408	4.19800
31-Jul	1-to-1	0.83651	0.79941	0.88726
	loc3	0.88693	0.87353	0.90000
	M-to-1 (entropy)	0.93509	0.92972	0.94583
	Avg. Conv. Length	13.71866	12.64815	14.84783
	Avg. Conv. Density	1.64763	1.51830	1.77452
	# Threads	50.00000	46.00000	54.00000
	Entropy	4.90776	4.76914	5.17706
Avg. All Sessions	1-to-1	0.74359	0.69245	0.80493
	loc3	0.87330	0.85220	0.89522
	M-to-1 (entropy)	0.87647	0.84806	0.90293
	Avg. Conv. Length	15.32323	13.76390	16.93643
	Avg. Conv. Density	1.86632	1.73753	2.00879
	# Threads	44.63333	40.40000	48.90000

Continued...

Session	Metric	Mean	Min	Max
	Entropy	4.39527	4.19973	4.61509

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX F: MAXIMUM ENTROPY CLASSIFICATION RESULTS

The following are the full results from maximum entropy classification over ##iphone, ##physics, and #python chat sessions. Two approaches were used: Same Annotator, Different Session – the training set was from a different session, but was annotated by the same person as the test set; and Same Session, Different Annotator – the training set was the same session as the test set, but was annotated by a different person. Results from both approaches are provided. The filename keys are as follows:
Same Annotator, Different Session: [chat topic]_[month]_[day of training session]-[annotator number]-[day of test session]-[annotator number]
Same Session, Different Annotator: [chat topic]_[month]_[day]-[training annotator]-[test annotator]

F.1 ##iphone, SAME ANNOTATOR, DIFFERENT SESSION

File	Accuracy	Precision	Recall	F-score
`iphone_07_17-1-18-1`	0.8915	0.8915	1.0000	0.9426
`iphone_07_17-1-19-1`	0.9126	0.9126	1.0000	0.9543
`iphone_07_17-1-21-1`	0.8300	0.8300	1.0000	0.9071
`iphone_07_17-1-22-1`	0.8055	0.8055	1.0000	0.8923
`iphone_07_17-1-23-1`	0.7786	0.7786	1.0000	0.8755
`iphone_07_17-1-24-1`	0.7255	0.7255	1.0000	0.8409
`iphone_07_17-1-28-1`	0.9578	0.9578	1.0000	0.9784
`iphone_07_17-1-29-1`	0.9366	0.9366	1.0000	0.9673
`iphone_07_17-1-31-1`	0.8056	0.8056	1.0000	0.8924
`iphone_07_17-2-18-2`	0.9492	0.9501	0.9990	0.9739
`iphone_07_17-2-19-2`	0.9701	0.9715	0.9986	0.9848
`iphone_07_17-2-21-2`	0.9055	0.9088	0.9960	0.9504
`iphone_07_17-2-22-2`	0.8440	0.8464	0.9964	0.9153

Continued...

File	Accuracy	Precision	Recall	F-score
iphone_07_17-2-23-2	0.6941	0.6941	1.0000	0.8194
iphone_07_17-2-24-2	0.7177	0.7194	0.9958	0.8354
iphone_07_17-2-28-2	0.9406	0.9413	0.9992	0.9694
iphone_07_17-2-29-2	0.9254	0.9263	0.9989	0.9612
iphone_07_17-2-31-2	0.8106	0.8133	0.9959	0.8954
iphone_07_17-3-18-3	0.8028	0.8028	1.0000	0.8906
iphone_07_17-3-19-3	0.8402	0.8402	1.0000	0.9132
iphone_07_17-3-21-3	0.8483	0.8483	1.0000	0.9179
iphone_07_17-3-22-3	0.7619	0.7619	1.0000	0.8648
iphone_07_17-3-23-3	0.7097	0.7097	1.0000	0.8302
iphone_07_17-3-24-3	0.7435	0.7436	0.9999	0.8529
iphone_07_17-3-28-3	0.9635	0.9635	1.0000	0.9814
iphone_07_17-3-29-3	0.8728	0.8728	1.0000	0.9321
iphone_07_17-3-31-3	0.7940	0.7940	1.0000	0.8852
iphone_07_18-1-17-1	0.9854	0.9854	1.0000	0.9927
iphone_07_18-1-19-1	0.9007	0.9149	0.9826	0.9475
iphone_07_18-1-21-1	0.8293	0.8348	0.9903	0.9059
iphone_07_18-1-22-1	0.8035	0.8125	0.9829	0.8896
iphone_07_18-1-23-1	0.7814	0.7812	0.9991	0.8768
iphone_07_18-1-24-1	0.7310	0.7336	0.9881	0.8420
iphone_07_18-1-28-1	0.9585	0.9598	0.9985	0.9788
iphone_07_18-1-29-1	0.9366	0.9434	0.9918	0.9670
iphone_07_18-1-31-1	0.8023	0.8050	0.9959	0.8903
iphone_07_18-2-17-2	0.9549	0.9549	1.0000	0.9769
iphone_07_18-2-19-2	0.9712	0.9715	0.9997	0.9854
iphone_07_18-2-21-2	0.9082	0.9082	1.0000	0.9519
iphone_07_18-2-22-2	0.8434	0.8458	0.9967	0.9151
iphone_07_18-2-23-2	0.6934	0.6939	0.9990	0.8189
iphone_07_18-2-24-2	0.7210	0.7205	0.9999	0.8375
iphone_07_18-2-28-2	0.9406	0.9406	1.0000	0.9694
iphone_07_18-2-29-2	0.9259	0.9259	1.0000	0.9615
iphone_07_18-2-31-2	0.8140	0.8140	1.0000	0.8974

Continued. . .

File	Accuracy	Precision	Recall	F-score
iphone_07_18-3-17-3	0.9607	0.9620	0.9985	0.9799
iphone_07_18-3-19-3	0.8368	0.8647	0.9553	0.9077
iphone_07_18-3-21-3	0.8381	0.8643	0.9598	0.9096
iphone_07_18-3-22-3	0.7666	0.7858	0.9537	0.8616
iphone_07_18-3-23-3	0.7268	0.7227	0.9980	0.8383
iphone_07_18-3-24-3	0.7485	0.7627	0.9607	0.8503
iphone_07_18-3-28-3	0.9614	0.9675	0.9933	0.9802
iphone_07_18-3-29-3	0.8631	0.8879	0.9649	0.9248
iphone_07_18-3-31-3	0.8056	0.8034	1.0000	0.8910
iphone_07_19-1-17-1	0.9854	0.9854	1.0000	0.9927
iphone_07_19-1-18-1	0.8913	0.8918	0.9993	0.9425
iphone_07_19-1-21-1	0.8298	0.8303	0.9991	0.9069
iphone_07_19-1-22-1	0.8056	0.8061	0.9990	0.8922
iphone_07_19-1-23-1	0.7800	0.7797	1.0000	0.8762
iphone_07_19-1-24-1	0.7278	0.7277	0.9984	0.8418
iphone_07_19-1-28-1	0.9564	0.9577	0.9985	0.9777
iphone_07_19-1-29-1	0.9366	0.9366	1.0000	0.9673
iphone_07_19-1-31-1	0.8073	0.8070	1.0000	0.8932
iphone_07_19-2-17-2	0.9549	0.9549	1.0000	0.9769
iphone_07_19-2-18-2	0.9501	0.9501	1.0000	0.9744
iphone_07_19-2-21-2	0.9087	0.9088	0.9997	0.9521
iphone_07_19-2-22-2	0.8463	0.8463	1.0000	0.9168
iphone_07_19-2-23-2	0.6948	0.6946	1.0000	0.8198
iphone_07_19-2-24-2	0.7193	0.7194	0.9994	0.8366
iphone_07_19-2-28-2	0.9399	0.9406	0.9992	0.9690
iphone_07_19-2-29-2	0.9259	0.9259	1.0000	0.9615
iphone_07_19-2-31-2	0.8140	0.8140	1.0000	0.8974
iphone_07_19-3-17-3	0.9578	0.9592	0.9985	0.9784
iphone_07_19-3-18-3	0.8106	0.8139	0.9906	0.8936
iphone_07_19-3-21-3	0.8410	0.8538	0.9805	0.9128
iphone_07_19-3-22-3	0.7700	0.7742	0.9856	0.8672
iphone_07_19-3-23-3	0.7104	0.7126	0.9920	0.8294

Continued...

File	Accuracy	Precision	Recall	F-score
iphone_07_19-3-24-3	0.7477	0.7541	0.9804	0.8525
iphone_07_19-3-28-3	0.9635	0.9662	0.9970	0.9814
iphone_07_19-3-29-3	0.8758	0.8797	0.9936	0.9332
iphone_07_19-3-31-3	0.8023	0.8007	1.0000	0.8893
iphone_07_21-1-17-1	0.9811	0.9854	0.9956	0.9904
iphone_07_21-1-18-1	0.8791	0.9050	0.9658	0.9344
iphone_07_21-1-19-1	0.8816	0.9165	0.9575	0.9366
iphone_07_21-1-22-1	0.7985	0.8205	0.9599	0.8847
iphone_07_21-1-23-1	0.7750	0.7794	0.9918	0.8728
iphone_07_21-1-24-1	0.7240	0.7348	0.9695	0.8360
iphone_07_21-1-28-1	0.9535	0.9603	0.9925	0.9761
iphone_07_21-1-29-1	0.9152	0.9450	0.9656	0.9552
iphone_07_21-1-31-1	0.7874	0.8020	0.9773	0.8810
iphone_07_21-2-17-2	0.9534	0.9548	0.9985	0.9762
iphone_07_21-2-18-2	0.9432	0.9514	0.9909	0.9707
iphone_07_21-2-19-2	0.9618	0.9745	0.9865	0.9805
iphone_07_21-2-22-2	0.8404	0.8489	0.9871	0.9128
iphone_07_21-2-23-2	0.6899	0.6928	0.9939	0.8165
iphone_07_21-2-24-2	0.7229	0.7264	0.9863	0.8366
iphone_07_21-2-28-2	0.9399	0.9425	0.9970	0.9690
iphone_07_21-2-29-2	0.9269	0.9299	0.9961	0.9619
iphone_07_21-2-31-2	0.8256	0.8246	0.9980	0.9030
iphone_07_21-3-17-3	0.9607	0.9620	0.9985	0.9799
iphone_07_21-3-18-3	0.8055	0.8122	0.9856	0.8906
iphone_07_21-3-19-3	0.8356	0.8484	0.9793	0.9092
iphone_07_21-3-22-3	0.7658	0.7722	0.9824	0.8647
iphone_07_21-3-23-3	0.7048	0.7092	0.9900	0.8264
iphone_07_21-3-24-3	0.7448	0.7526	0.9783	0.8508
iphone_07_21-3-28-3	0.9607	0.9641	0.9963	0.9799
iphone_07_21-3-29-3	0.8687	0.8800	0.9836	0.9289
iphone_07_21-3-31-3	0.7990	0.8000	0.9958	0.8872
iphone_07_22-1-17-1	0.9854	0.9854	1.0000	0.9927

Continued…

File	Accuracy	Precision	Recall	F-score
iphone_07_22-1-18-1	0.8921	0.8979	0.9917	0.9425
iphone_07_22-1-19-1	0.9009	0.9157	0.9818	0.9476
iphone_07_22-1-21-1	0.8325	0.8367	0.9918	0.9077
iphone_07_22-1-23-1	0.7750	0.7818	0.9863	0.8722
iphone_07_22-1-24-1	0.7281	0.7341	0.9805	0.8396
iphone_07_22-1-28-1	0.9542	0.9603	0.9933	0.9765
iphone_07_22-1-29-1	0.9346	0.9410	0.9924	0.9660
iphone_07_22-1-31-1	0.8056	0.8056	1.0000	0.8924
iphone_07_22-2-17-2	0.9549	0.9549	1.0000	0.9769
iphone_07_22-2-18-2	0.9400	0.9530	0.9855	0.9690
iphone_07_22-2-19-2	0.9589	0.9728	0.9853	0.9790
iphone_07_22-2-21-2	0.9060	0.9120	0.9922	0.9504
iphone_07_22-2-23-2	0.6955	0.6954	0.9990	0.8200
iphone_07_22-2-24-2	0.7219	0.7234	0.9928	0.8370
iphone_07_22-2-28-2	0.9421	0.9420	1.0000	0.9701
iphone_07_22-2-29-2	0.9228	0.9301	0.9912	0.9597
iphone_07_22-2-31-2	0.8156	0.8153	1.0000	0.8983
iphone_07_22-3-17-3	0.9520	0.9589	0.9924	0.9754
iphone_07_22-3-18-3	0.8140	0.8267	0.9720	0.8935
iphone_07_22-3-19-3	0.8292	0.8604	0.9509	0.9034
iphone_07_22-3-21-3	0.8366	0.8633	0.9592	0.9088
iphone_07_22-3-23-3	0.7168	0.7251	0.9680	0.8291
iphone_07_22-3-24-3	0.7567	0.7718	0.9552	0.8538
iphone_07_22-3-28-3	0.9499	0.9671	0.9814	0.9742
iphone_07_22-3-29-3	0.8707	0.8939	0.9666	0.9288
iphone_07_22-3-31-3	0.8056	0.8044	0.9979	0.8908
iphone_07_23-1-17-1	0.9723	0.9853	0.9867	0.9860
iphone_07_23-1-18-1	0.8626	0.9173	0.9297	0.9234
iphone_07_23-1-19-1	0.8281	0.9218	0.8868	0.9040
iphone_07_23-1-21-1	0.8018	0.8510	0.9229	0.8854
iphone_07_23-1-22-1	0.7735	0.8278	0.9076	0.8659
iphone_07_23-1-24-1	0.7330	0.7561	0.9330	0.8353

Continued...

File	Accuracy	Precision	Recall	F-score
iphone_07_23-1-28-1	0.9328	0.9615	0.9686	0.9650
iphone_07_23-1-29-1	0.8702	0.9530	0.9062	0.9290
iphone_07_23-1-31-1	0.7990	0.8095	0.9814	0.8872
iphone_07_23-2-17-2	0.8690	0.9506	0.9101	0.9299
iphone_07_23-2-18-2	0.7638	0.9580	0.7859	0.8634
iphone_07_23-2-19-2	0.7659	0.9733	0.7804	0.8663
iphone_07_23-2-21-2	0.8011	0.9354	0.8389	0.8845
iphone_07_23-2-22-2	0.7745	0.8883	0.8389	0.8629
iphone_07_23-2-24-2	0.7044	0.7601	0.8606	0.8072
iphone_07_23-2-28-2	0.8398	0.9453	0.8806	0.9118
iphone_07_23-2-29-2	0.7936	0.9373	0.8328	0.8819
iphone_07_23-2-31-2	0.8173	0.8493	0.9429	0.8936
iphone_07_23-3-17-3	0.8923	0.9563	0.9302	0.9431
iphone_07_23-3-18-3	0.7630	0.8589	0.8434	0.8511
iphone_07_23-3-19-3	0.7334	0.8837	0.7862	0.8321
iphone_07_23-3-21-3	0.7465	0.8838	0.8074	0.8439
iphone_07_23-3-22-3	0.7084	0.8171	0.7953	0.8060
iphone_07_23-3-24-3	0.7028	0.7898	0.8181	0.8037
iphone_07_23-3-28-3	0.8920	0.9709	0.9154	0.9423
iphone_07_23-3-29-3	0.7874	0.9115	0.8378	0.8731
iphone_07_23-3-31-3	0.8173	0.8370	0.9561	0.8926
iphone_07_24-1-17-1	0.9636	0.9851	0.9778	0.9815
iphone_07_24-1-18-1	0.8729	0.9147	0.9455	0.9299
iphone_07_24-1-19-1	0.8484	0.9213	0.9118	0.9165
iphone_07_24-1-21-1	0.8196	0.8573	0.9390	0.8963
iphone_07_24-1-22-1	0.7825	0.8332	0.9127	0.8711
iphone_07_24-1-23-1	0.7928	0.7953	0.9881	0.8813
iphone_07_24-1-28-1	0.9456	0.9640	0.9798	0.9719
iphone_07_24-1-29-1	0.9142	0.9567	0.9514	0.9540
iphone_07_24-1-31-1	0.7924	0.8093	0.9711	0.8828
iphone_07_24-2-17-2	0.9360	0.9540	0.9802	0.9669
iphone_07_24-2-18-2	0.8956	0.9543	0.9349	0.9445

Continued. . .

File	Accuracy	Precision	Recall	F-score
iphone_07_24-2-19-2	0.8859	0.9746	0.9062	0.9392
iphone_07_24-2-21-2	0.8783	0.9309	0.9354	0.9331
iphone_07_24-2-22-2	0.8160	0.8685	0.9222	0.8945
iphone_07_24-2-23-2	0.6977	0.7113	0.9499	0.8135
iphone_07_24-2-28-2	0.9263	0.9469	0.9764	0.9614
iphone_07_24-2-29-2	0.9024	0.9417	0.9536	0.9476
iphone_07_24-2-31-2	0.8040	0.8207	0.9714	0.8897
iphone_07_24-3-17-3	0.9534	0.9590	0.9939	0.9762
iphone_07_24-3-18-3	0.8139	0.8215	0.9814	0.8943
iphone_07_24-3-19-3	0.8362	0.8571	0.9661	0.9084
iphone_07_24-3-21-3	0.8403	0.8624	0.9658	0.9112
iphone_07_24-3-22-3	0.7748	0.7860	0.9680	0.8675
iphone_07_24-3-23-3	0.7161	0.7168	0.9920	0.8322
iphone_07_24-3-28-3	0.9578	0.9667	0.9903	0.9784
iphone_07_24-3-29-3	0.8728	0.8878	0.9778	0.9306
iphone_07_24-3-31-3	0.8040	0.8041	0.9958	0.8897
iphone_07_28-1-17-1	0.9767	0.9853	0.9911	0.9882
iphone_07_28-1-18-1	0.8924	0.8937	0.9980	0.9430
iphone_07_28-1-19-1	0.9097	0.9131	0.9958	0.9526
iphone_07_28-1-21-1	0.8286	0.8316	0.9950	0.9060
iphone_07_28-1-22-1	0.8047	0.8085	0.9925	0.8911
iphone_07_28-1-23-1	0.7800	0.7805	0.9982	0.8760
iphone_07_28-1-24-1	0.7314	0.7308	0.9968	0.8434
iphone_07_28-1-29-1	0.9361	0.9393	0.9962	0.9669
iphone_07_28-1-31-1	0.8073	0.8070	1.0000	0.8932
iphone_07_28-2-17-2	0.9432	0.9557	0.9863	0.9707
iphone_07_28-2-18-2	0.9460	0.9515	0.9938	0.9722
iphone_07_28-2-19-2	0.9640	0.9744	0.9889	0.9816
iphone_07_28-2-21-2	0.9041	0.9114	0.9906	0.9494
iphone_07_28-2-22-2	0.8431	0.8521	0.9856	0.9140
iphone_07_28-2-23-2	0.6962	0.6961	0.9980	0.8202
iphone_07_28-2-24-2	0.7195	0.7237	0.9866	0.8349

Continued…

File	Accuracy	Precision	Recall	F-score
iphone_07_28-2-29-2	0.9228	0.9292	0.9923	0.9597
iphone_07_28-2-31-2	0.8173	0.8188	0.9959	0.8987
iphone_07_28-3-17-3	0.9476	0.9588	0.9879	0.9731
iphone_07_28-3-18-3	0.8068	0.8076	0.9968	0.8923
iphone_07_28-3-19-3	0.8404	0.8430	0.9954	0.9129
iphone_07_28-3-21-3	0.8427	0.8488	0.9911	0.9145
iphone_07_28-3-22-3	0.7615	0.7667	0.9875	0.8632
iphone_07_28-3-23-3	0.7147	0.7133	1.0000	0.8326
iphone_07_28-3-24-3	0.7487	0.7496	0.9943	0.8547
iphone_07_28-3-29-3	0.8728	0.8770	0.9936	0.9316
iphone_07_28-3-31-3	0.8007	0.7993	1.0000	0.8885
iphone_07_29-1-17-1	0.9854	0.9854	1.0000	0.9927
iphone_07_29-1-18-1	0.8913	0.8918	0.9993	0.9425
iphone_07_29-1-19-1	0.9095	0.9128	0.9960	0.9526
iphone_07_29-1-21-1	0.8288	0.8300	0.9982	0.9064
iphone_07_29-1-22-1	0.8062	0.8070	0.9981	0.8924
iphone_07_29-1-23-1	0.7793	0.7791	1.0000	0.8758
iphone_07_29-1-24-1	0.7272	0.7270	0.9991	0.8416
iphone_07_29-1-28-1	0.9578	0.9578	1.0000	0.9784
iphone_07_29-1-31-1	0.8056	0.8056	1.0000	0.8924
iphone_07_29-2-17-2	0.9549	0.9549	1.0000	0.9769
iphone_07_29-2-18-2	0.9492	0.9502	0.9988	0.9739
iphone_07_29-2-19-2	0.9703	0.9738	0.9962	0.9849
iphone_07_29-2-21-2	0.9070	0.9097	0.9965	0.9511
iphone_07_29-2-22-2	0.8437	0.8473	0.9945	0.9150
iphone_07_29-2-23-2	0.6955	0.6951	1.0000	0.8201
iphone_07_29-2-24-2	0.7195	0.7211	0.9945	0.8360
iphone_07_29-2-28-2	0.9413	0.9419	0.9992	0.9697
iphone_07_29-2-31-2	0.8206	0.8194	1.0000	0.9007
iphone_07_29-3-17-3	0.9374	0.9583	0.9772	0.9677
iphone_07_29-3-18-3	0.8004	0.8133	0.9752	0.8869
iphone_07_29-3-19-3	0.8255	0.8527	0.9577	0.9022

Continued…

File	Accuracy	Precision	Recall	F-score
iphone_07_29-3-21-3	0.8366	0.8615	0.9621	0.9090
iphone_07_29-3-22-3	0.7568	0.7769	0.9550	0.8568
iphone_07_29-3-23-3	0.6984	0.7109	0.9690	0.8201
iphone_07_29-3-24-3	0.7328	0.7565	0.9447	0.8402
iphone_07_29-3-28-3	0.9413	0.9647	0.9748	0.9697
iphone_07_29-3-31-3	0.8090	0.8212	0.9707	0.8897
iphone_07_31-1-17-1	0.9229	0.9875	0.9335	0.9598
iphone_07_31-1-18-1	0.8296	0.8925	0.9196	0.9058
iphone_07_31-1-19-1	0.8521	0.9098	0.9302	0.9199
iphone_07_31-1-21-1	0.8042	0.8395	0.9449	0.8890
iphone_07_31-1-22-1	0.7543	0.8120	0.9043	0.8557
iphone_07_31-1-23-1	0.7204	0.7693	0.9152	0.8360
iphone_07_31-1-24-1	0.6978	0.7267	0.9351	0.8179
iphone_07_31-1-28-1	0.8984	0.9572	0.9358	0.9464
iphone_07_31-1-29-1	0.8820	0.9377	0.9362	0.9369
iphone_07_31-2-17-2	0.9025	0.9538	0.9436	0.9487
iphone_07_31-2-18-2	0.8587	0.9518	0.8967	0.9235
iphone_07_31-2-19-2	0.8870	0.9737	0.9083	0.9398
iphone_07_31-2-21-2	0.8420	0.9174	0.9078	0.9125
iphone_07_31-2-22-2	0.7660	0.8550	0.8712	0.8630
iphone_07_31-2-23-2	0.6863	0.6959	0.9734	0.8116
iphone_07_31-2-24-2	0.6819	0.7184	0.9172	0.8057
iphone_07_31-2-28-2	0.8970	0.9439	0.9468	0.9453
iphone_07_31-2-29-2	0.8615	0.9341	0.9150	0.9244
iphone_07_31-3-17-3	0.8763	0.9614	0.9074	0.9336
iphone_07_31-3-18-3	0.7404	0.8152	0.8750	0.8440
iphone_07_31-3-19-3	0.7671	0.8467	0.8826	0.8643
iphone_07_31-3-21-3	0.7991	0.8719	0.8947	0.8831
iphone_07_31-3-22-3	0.6824	0.7700	0.8314	0.7995
iphone_07_31-3-23-3	0.7076	0.7191	0.9650	0.8241
iphone_07_31-3-24-3	0.6943	0.7491	0.8854	0.8116
iphone_07_31-3-28-3	0.8941	0.9644	0.9243	0.9439

Continued...

File	Accuracy	Precision	Recall	F-score
iphone_07_31-3-29-3	0.8160	0.8910	0.8993	0.8951
Min	0.6819	0.6928	0.7804	0.7995
Max	0.9854	0.9875	1.0000	0.9927
Avg	0.8405	0.8618	0.9693	0.9093
Std Dev	0.0851	0.0864	0.0456	0.0522

F.2 ##iphone, SAME SESSION, DIFFERENT ANNOTATOR

File	Accuracy	Precision	Recall	F-score
iphone_07_17-1-2	0.9549	0.9549	1.0000	0.9769
iphone_07_17-1-3	0.9592	0.9592	1.0000	0.9792
iphone_07_17-2-1	0.9869	0.9869	1.0000	0.9934
iphone_07_17-2-3	0.9607	0.9606	1.0000	0.9799
iphone_07_17-3-1	0.9854	0.9854	1.0000	0.9927
iphone_07_17-3-2	0.9549	0.9549	1.0000	0.9769
iphone_07_18-1-2	0.9404	0.9536	0.9852	0.9691
iphone_07_18-1-3	0.8087	0.8115	0.9922	0.8928
iphone_07_18-2-1	0.8915	0.8915	1.0000	0.9426
iphone_07_18-2-3	0.8028	0.8028	1.0000	0.8906
iphone_07_18-3-1	0.8868	0.9121	0.9662	0.9383
iphone_07_18-3-2	0.9169	0.9591	0.9533	0.9562
iphone_07_19-1-2	0.9714	0.9728	0.9984	0.9855
iphone_07_19-1-3	0.8410	0.8416	0.9987	0.9135
iphone_07_19-2-1	0.9138	0.9138	0.9998	0.9549
iphone_07_19-2-3	0.8418	0.8415	1.0000	0.9139
iphone_07_19-3-1	0.8975	0.9166	0.9766	0.9456
iphone_07_19-3-2	0.9509	0.9744	0.9751	0.9747
iphone_07_21-1-2	0.8955	0.9201	0.9692	0.9440
iphone_07_21-1-3	0.8507	0.8654	0.9759	0.9173

Continued. . .

File	Accuracy	Precision	Recall	F-score
iphone_07_21-2-1	0.8381	0.8388	0.9965	0.9108
iphone_07_21-2-3	0.8539	0.8560	0.9951	0.9204
iphone_07_21-3-1	0.8442	0.8467	0.9918	0.9135
iphone_07_21-3-2	0.9058	0.9186	0.9834	0.9499
iphone_07_22-1-2	0.8365	0.8503	0.9791	0.9102
iphone_07_22-1-3	0.7717	0.7738	0.9895	0.8685
iphone_07_22-2-1	0.8033	0.8097	0.9881	0.8900
iphone_07_22-2-3	0.7627	0.7668	0.9894	0.8640
iphone_07_22-3-1	0.8019	0.8302	0.9480	0.8852
iphone_07_22-3-2	0.8249	0.8648	0.9401	0.9009
iphone_07_23-1-2	0.7019	0.7098	0.9652	0.8180
iphone_07_23-1-3	0.7374	0.7368	0.9800	0.8412
iphone_07_23-2-1	0.7374	0.8068	0.8715	0.8379
iphone_07_23-2-3	0.7140	0.7519	0.8910	0.8156
iphone_07_23-3-1	0.7793	0.8129	0.9307	0.8678
iphone_07_23-3-2	0.7104	0.7269	0.9335	0.8174
iphone_07_24-1-2	0.7277	0.7485	0.9359	0.8317
iphone_07_24-1-3	0.7470	0.7728	0.9345	0.8460
iphone_07_24-2-1	0.7304	0.7585	0.9220	0.8323
iphone_07_24-2-3	0.7452	0.7771	0.9216	0.8432
iphone_07_24-3-1	0.7344	0.7460	0.9612	0.8400
iphone_07_24-3-2	0.7310	0.7407	0.9630	0.8374
iphone_07_28-1-2	0.9428	0.9427	1.0000	0.9705
iphone_07_28-1-3	0.9657	0.9656	1.0000	0.9825
iphone_07_28-2-1	0.9557	0.9590	0.9963	0.9773
iphone_07_28-2-3	0.9614	0.9648	0.9963	0.9803
iphone_07_28-3-1	0.9607	0.9605	1.0000	0.9799
iphone_07_28-3-2	0.9435	0.9433	1.0000	0.9708
iphone_07_29-1-2	0.9269	0.9282	0.9983	0.9620
iphone_07_29-1-3	0.8758	0.8758	0.9994	0.9336
iphone_07_29-2-1	0.9382	0.9394	0.9984	0.9680
iphone_07_29-2-3	0.8763	0.8763	0.9994	0.9338

Continued. . .

File	Accuracy	Precision	Recall	F-score
iphone_07_29-3-1	0.9244	0.9489	0.9716	0.9601
iphone_07_29-3-2	0.9136	0.9377	0.9713	0.9542
iphone_07_31-1-2	0.8306	0.8647	0.9388	0.9002
iphone_07_31-1-3	0.8339	0.8553	0.9519	0.9010
iphone_07_31-2-1	0.8090	0.8426	0.9381	0.8878
iphone_07_31-2-3	0.8439	0.8556	0.9665	0.9077
iphone_07_31-3-1	0.8339	0.8681	0.9361	0.9008
iphone_07_31-3-2	0.8654	0.8910	0.9510	0.9200
Min	0.7019	0.7098	0.8715	0.8156
Max	0.9869	0.9869	1.0000	0.9934
Avg	0.8575	0.8707	0.9736	0.9178
Std Dev	0.0842	0.0792	0.0302	0.0530

F.3 ##physics, SAME ANNOTATOR, DIFFERENT SESSION

File	Accuracy	Precision	Recall	F-score
physics_07_17-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_17-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_17-1-21-1	0.9518	0.9518	1.0000	0.9753
physics_07_17-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_17-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_17-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_17-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_17-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_17-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_17-2-18-2	0.9541	0.9949	0.9588	0.9765
physics_07_17-2-19-2	0.9027	0.9076	0.9933	0.9486
physics_07_17-2-21-2	0.9391	0.9665	0.9705	0.9685
physics_07_17-2-22-2	0.9718	0.9818	0.9895	0.9857

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_17-2-23-2	0.9669	0.9759	0.9906	0.9832
physics_07_17-2-24-2	0.7818	0.8194	0.9409	0.8760
physics_07_17-2-28-2	0.6409	0.6515	0.9675	0.7786
physics_07_17-2-29-2	0.8685	0.9128	0.9457	0.9290
physics_07_17-2-31-2	0.9586	0.9642	0.9939	0.9788
physics_07_17-3-18-3	1.0000	1.0000	1.0000	1.0000
physics_07_17-3-19-3	0.9393	0.9393	1.0000	0.9687
physics_07_17-3-21-3	0.9409	0.9409	1.0000	0.9696
physics_07_17-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_17-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_17-3-24-3	0.8645	0.8645	1.0000	0.9273
physics_07_17-3-28-3	0.6845	0.6845	1.0000	0.8127
physics_07_17-3-29-3	0.8387	0.8387	1.0000	0.9123
physics_07_17-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_18-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_18-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_18-1-21-1	0.9518	0.9518	1.0000	0.9753
physics_07_18-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_18-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_18-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_18-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_18-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_18-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_18-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_18-2-19-2	0.9031	0.9031	1.0000	0.9491
physics_07_18-2-21-2	0.9651	0.9651	1.0000	0.9822
physics_07_18-2-22-2	0.9820	0.9820	1.0000	0.9909
physics_07_18-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_18-2-24-2	0.8188	0.8188	1.0000	0.9004
physics_07_18-2-28-2	0.6528	0.6528	1.0000	0.7899
physics_07_18-2-29-2	0.9093	0.9093	1.0000	0.9525
physics_07_18-2-31-2	0.9625	0.9625	1.0000	0.9809

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_18-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_18-3-19-3	0.9393	0.9393	1.0000	0.9687
physics_07_18-3-21-3	0.9409	0.9409	1.0000	0.9696
physics_07_18-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_18-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_18-3-24-3	0.8645	0.8645	1.0000	0.9273
physics_07_18-3-28-3	0.6845	0.6845	1.0000	0.8127
physics_07_18-3-29-3	0.8387	0.8387	1.0000	0.9123
physics_07_18-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_19-1-17-1	0.9750	0.9873	0.9873	0.9873
physics_07_19-1-18-1	0.9721	0.9950	0.9769	0.9859
physics_07_19-1-21-1	0.9304	0.9515	0.9767	0.9639
physics_07_19-1-22-1	0.9961	1.0000	0.9961	0.9981
physics_07_19-1-23-1	0.9963	1.0000	0.9963	0.9982
physics_07_19-1-24-1	0.8551	0.8948	0.9494	0.9213
physics_07_19-1-28-1	0.8751	0.9154	0.9503	0.9325
physics_07_19-1-29-1	0.9589	0.9869	0.9713	0.9790
physics_07_19-1-31-1	0.9763	0.9762	1.0000	0.9880
physics_07_19-2-17-2	0.9625	0.9618	1.0000	0.9805
physics_07_19-2-18-2	0.9525	0.9949	0.9572	0.9757
physics_07_19-2-21-2	0.9379	0.9684	0.9672	0.9678
physics_07_19-2-22-2	0.9782	0.9820	0.9961	0.9890
physics_07_19-2-23-2	0.9651	0.9758	0.9887	0.9822
physics_07_19-2-24-2	0.7818	0.8362	0.9122	0.8726
physics_07_19-2-28-2	0.6460	0.6658	0.9190	0.7722
physics_07_19-2-29-2	0.8800	0.9218	0.9485	0.9350
physics_07_19-2-31-2	0.9625	0.9644	0.9980	0.9809
physics_07_19-3-17-3	0.9875	1.0000	0.9875	0.9937
physics_07_19-3-18-3	0.9984	1.0000	0.9984	0.9992
physics_07_19-3-21-3	0.9282	0.9405	0.9861	0.9628
physics_07_19-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_19-3-23-3	0.9779	0.9779	1.0000	0.9888

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_19-3-24-3	0.8395	0.8650	0.9650	0.9123
physics_07_19-3-28-3	0.6824	0.6899	0.9738	0.8076
physics_07_19-3-29-3	0.8395	0.8449	0.9904	0.9119
physics_07_19-3-31-3	0.9310	0.9307	1.0000	0.9641
physics_07_21-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_21-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_21-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_21-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_21-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_21-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_21-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_21-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_21-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_21-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_21-2-18-2	0.9934	0.9951	0.9984	0.9967
physics_07_21-2-19-2	0.9047	0.9046	1.0000	0.9499
physics_07_21-2-22-2	0.9820	0.9820	1.0000	0.9909
physics_07_21-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_21-2-24-2	0.8178	0.8187	0.9985	0.8997
physics_07_21-2-28-2	0.6519	0.6525	0.9986	0.7893
physics_07_21-2-29-2	0.9098	0.9100	0.9997	0.9528
physics_07_21-2-31-2	0.9625	0.9625	1.0000	0.9809
physics_07_21-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_21-3-18-3	0.9869	1.0000	0.9869	0.9934
physics_07_21-3-19-3	0.9389	0.9393	0.9996	0.9685
physics_07_21-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_21-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_21-3-24-3	0.8633	0.8644	0.9984	0.9266
physics_07_21-3-28-3	0.6869	0.6864	0.9991	0.8137
physics_07_21-3-29-3	0.8375	0.8392	0.9973	0.9114
physics_07_21-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_22-1-17-1	0.9875	0.9875	1.0000	0.9937

Continued...

File	Accuracy	Precision	Recall	F-score
physics_07_22-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_22-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_22-1-21-1	0.9518	0.9518	1.0000	0.9753
physics_07_22-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_22-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_22-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_22-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_22-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_22-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_22-2-18-2	0.9951	0.9951	1.0000	0.9975
physics_07_22-2-19-2	0.9027	0.9031	0.9996	0.9489
physics_07_22-2-21-2	0.9647	0.9651	0.9996	0.9820
physics_07_22-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_22-2-24-2	0.8184	0.8189	0.9991	0.9001
physics_07_22-2-28-2	0.6522	0.6526	0.9991	0.7895
physics_07_22-2-29-2	0.9076	0.9092	0.9981	0.9516
physics_07_22-2-31-2	0.9625	0.9625	1.0000	0.9809
physics_07_22-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_22-3-18-3	1.0000	1.0000	1.0000	1.0000
physics_07_22-3-19-3	0.9393	0.9393	1.0000	0.9687
physics_07_22-3-21-3	0.9409	0.9409	1.0000	0.9696
physics_07_22-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_22-3-24-3	0.8645	0.8645	1.0000	0.9273
physics_07_22-3-28-3	0.6845	0.6845	1.0000	0.8127
physics_07_22-3-29-3	0.8387	0.8387	1.0000	0.9123
physics_07_22-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_23-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_23-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_23-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_23-1-21-1	0.9518	0.9518	1.0000	0.9753
physics_07_23-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_23-1-24-1	0.8936	0.8936	1.0000	0.9438

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_23-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_23-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_23-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_23-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_23-2-18-2	0.9967	0.9967	1.0000	0.9984
physics_07_23-2-19-2	0.9035	0.9035	1.0000	0.9493
physics_07_23-2-21-2	0.9649	0.9653	0.9996	0.9821
physics_07_23-2-22-2	0.9820	0.9820	1.0000	0.9909
physics_07_23-2-24-2	0.8188	0.8188	1.0000	0.9004
physics_07_23-2-28-2	0.6549	0.6542	1.0000	0.7910
physics_07_23-2-29-2	0.9083	0.9093	0.9989	0.9520
physics_07_23-2-31-2	0.9606	0.9625	0.9980	0.9799
physics_07_23-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_23-3-18-3	1.0000	1.0000	1.0000	1.0000
physics_07_23-3-19-3	0.9397	0.9397	1.0000	0.9689
physics_07_23-3-21-3	0.9393	0.9408	0.9983	0.9687
physics_07_23-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_23-3-24-3	0.8645	0.8645	1.0000	0.9273
physics_07_23-3-28-3	0.6863	0.6857	1.0000	0.8136
physics_07_23-3-29-3	0.8380	0.8386	0.9991	0.9118
physics_07_23-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_24-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_24-1-18-1	0.9836	0.9967	0.9868	0.9917
physics_07_24-1-19-1	0.9240	0.9243	0.9996	0.9604
physics_07_24-1-21-1	0.9500	0.9524	0.9972	0.9743
physics_07_24-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_24-1-23-1	0.9890	1.0000	0.9890	0.9945
physics_07_24-1-28-1	0.9110	0.9134	0.9964	0.9531
physics_07_24-1-29-1	0.9822	0.9874	0.9947	0.9910
physics_07_24-1-31-1	0.9842	0.9879	0.9959	0.9919
physics_07_24-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_24-2-18-2	0.9443	0.9983	0.9456	0.9712

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_24-2-19-2	0.8995	0.9094	0.9871	0.9466
physics_07_24-2-21-2	0.9445	0.9665	0.9764	0.9714
physics_07_24-2-22-2	0.9795	0.9820	0.9974	0.9896
physics_07_24-2-23-2	0.9835	0.9869	0.9962	0.9916
physics_07_24-2-28-2	0.6785	0.6729	0.9876	0.8004
physics_07_24-2-29-2	0.8830	0.9109	0.9658	0.9376
physics_07_24-2-31-2	0.9566	0.9775	0.9775	0.9775
physics_07_24-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_24-3-18-3	0.9967	1.0000	0.9967	0.9984
physics_07_24-3-19-3	0.9397	0.9407	0.9987	0.9689
physics_07_24-3-21-3	0.9373	0.9412	0.9955	0.9676
physics_07_24-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_24-3-23-3	0.9853	0.9870	0.9981	0.9925
physics_07_24-3-28-3	0.6944	0.6914	0.9996	0.8174
physics_07_24-3-29-3	0.8342	0.8382	0.9943	0.9096
physics_07_24-3-31-3	0.9250	0.9303	0.9936	0.9609
physics_07_28-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_28-1-18-1	0.9951	0.9967	0.9984	0.9975
physics_07_28-1-19-1	0.9240	0.9246	0.9991	0.9604
physics_07_28-1-21-1	0.9484	0.9524	0.9956	0.9735
physics_07_28-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_28-1-23-1	0.9890	1.0000	0.9890	0.9945
physics_07_28-1-24-1	0.8954	0.8954	0.9998	0.9447
physics_07_28-1-29-1	0.9812	0.9874	0.9937	0.9905
physics_07_28-1-31-1	0.9684	0.9742	0.9939	0.9839
physics_07_28-2-17-2	0.8125	0.9618	0.8344	0.8936
physics_07_28-2-18-2	0.7721	0.9979	0.7727	0.8709
physics_07_28-2-19-2	0.8042	0.9276	0.8495	0.8868
physics_07_28-2-21-2	0.8076	0.9809	0.8165	0.8912
physics_07_28-2-22-2	0.8549	0.9880	0.8627	0.9211
physics_07_28-2-23-2	0.8658	0.9935	0.8682	0.9266
physics_07_28-2-24-2	0.7067	0.8782	0.7452	0.8062

Continued...

File	Accuracy	Precision	Recall	F-score
physics_07_28-2-29-2	0.7458	0.9233	0.7857	0.8490
physics_07_28-2-31-2	0.8955	0.9844	0.9057	0.9434
physics_07_28-3-17-3	0.9625	1.0000	0.9625	0.9809
physics_07_28-3-18-3	0.8721	1.0000	0.8721	0.9317
physics_07_28-3-19-3	0.9015	0.9564	0.9379	0.9470
physics_07_28-3-21-3	0.8818	0.9497	0.9233	0.9363
physics_07_28-3-22-3	0.9409	1.0000	0.9409	0.9696
physics_07_28-3-23-3	0.9577	0.9885	0.9680	0.9782
physics_07_28-3-24-3	0.7610	0.8832	0.8338	0.8578
physics_07_28-3-29-3	0.8137	0.8676	0.9179	0.8920
physics_07_28-3-31-3	0.9250	0.9519	0.9681	0.9599
physics_07_29-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_29-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_29-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_29-1-21-1	0.9510	0.9518	0.9992	0.9749
physics_07_29-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_29-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_29-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_29-1-28-1	0.9077	0.9082	0.9993	0.9516
physics_07_29-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_29-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_29-2-18-2	0.9951	0.9951	1.0000	0.9975
physics_07_29-2-19-2	0.9031	0.9051	0.9973	0.9490
physics_07_29-2-21-2	0.9637	0.9656	0.9979	0.9815
physics_07_29-2-22-2	0.9807	0.9820	0.9987	0.9903
physics_07_29-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_29-2-24-2	0.8163	0.8209	0.9921	0.8984
physics_07_29-2-28-2	0.6528	0.6558	0.9854	0.7875
physics_07_29-2-31-2	0.9625	0.9625	1.0000	0.9809
physics_07_29-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_29-3-18-3	0.9574	1.0000	0.9574	0.9782
physics_07_29-3-19-3	0.9349	0.9470	0.9859	0.9660

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_29-3-21-3	0.9214	0.9427	0.9758	0.9589
physics_07_29-3-22-3	0.9859	1.0000	0.9859	0.9929
physics_07_29-3-23-3	0.9688	0.9777	0.9906	0.9841
physics_07_29-3-24-3	0.8232	0.8685	0.9374	0.9016
physics_07_29-3-28-3	0.6958	0.7077	0.9467	0.8099
physics_07_29-3-31-3	0.9310	0.9307	1.0000	0.9641
physics_07_31-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_31-1-18-1	0.9836	0.9950	0.9885	0.9917
physics_07_31-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_31-1-21-1	0.9506	0.9519	0.9985	0.9747
physics_07_31-1-22-1	0.9987	1.0000	0.9987	0.9994
physics_07_31-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_31-1-24-1	0.8932	0.8956	0.9966	0.9434
physics_07_31-1-28-1	0.9032	0.9093	0.9924	0.9490
physics_07_31-1-29-1	0.9860	0.9872	0.9987	0.9929
physics_07_31-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_31-2-18-2	0.9770	0.9950	0.9819	0.9884
physics_07_31-2-19-2	0.9055	0.9076	0.9969	0.9501
physics_07_31-2-21-2	0.9587	0.9655	0.9927	0.9789
physics_07_31-2-22-2	0.9782	0.9820	0.9961	0.9890
physics_07_31-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_31-2-24-2	0.8095	0.8227	0.9780	0.8937
physics_07_31-2-28-2	0.6430	0.6549	0.9579	0.7779
physics_07_31-2-29-2	0.9058	0.9123	0.9917	0.9504
physics_07_31-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_31-3-18-3	0.9590	1.0000	0.9590	0.9791
physics_07_31-3-19-3	0.9304	0.9489	0.9786	0.9635
physics_07_31-3-21-3	0.9292	0.9437	0.9835	0.9632
physics_07_31-3-22-3	0.9884	1.0000	0.9884	0.9942
physics_07_31-3-23-3	0.9743	0.9779	0.9962	0.9870
physics_07_31-3-24-3	0.8306	0.8668	0.9500	0.9065
physics_07_31-3-28-3	0.6713	0.6947	0.9275	0.7944

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_31-3-29-3	0.8325	0.8439	0.9818	0.9077
Min	0.6409	0.6515	0.7452	0.7722
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.9202	0.9322	0.9852	0.9556
Std Dev	0.0881	0.0840	0.0370	0.0532

F.4 ##physics, SAME SESSION, DIFFERENT ANNOTATOR

File	Accuracy	Precision	Recall	F-score
physics_07_17-1-2	0.9438	0.9438	1.0000	0.9711
physics_07_17-1-3	1.0000	1.0000	1.0000	1.0000
physics_07_17-2-1	0.9750	0.9873	0.9873	0.9873
physics_07_17-2-3	0.9875	1.0000	0.9875	0.9937
physics_07_17-3-1	0.9875	0.9875	1.0000	0.9937
physics_07_17-3-2	0.9438	0.9438	1.0000	0.9711
physics_07_18-1-2	0.9951	0.9951	1.0000	0.9975
physics_07_18-1-3	1.0000	1.0000	1.0000	1.0000
physics_07_18-2-1	0.9951	0.9951	1.0000	0.9975
physics_07_18-2-3	1.0000	1.0000	1.0000	1.0000
physics_07_18-3-1	0.9951	0.9951	1.0000	0.9975
physics_07_18-3-2	0.9951	0.9951	1.0000	0.9975
physics_07_19-1-2	0.9007	0.9075	0.9911	0.9474
physics_07_19-1-3	0.9361	0.9437	0.9910	0.9668
physics_07_19-2-1	0.9168	0.9293	0.9847	0.9562
physics_07_19-2-3	0.9389	0.9490	0.9880	0.9681
physics_07_19-3-1	0.9228	0.9259	0.9961	0.9597
physics_07_19-3-2	0.9103	0.9097	1.0000	0.9527
physics_07_21-1-2	0.9651	0.9651	1.0000	0.9822
physics_07_21-1-3	0.9409	0.9409	1.0000	0.9696

Continued…

File	Accuracy	Precision	Recall	F-score
physics_07_21-2-1	0.9516	0.9518	0.9998	0.9752
physics_07_21-2-3	0.9407	0.9409	0.9998	0.9695
physics_07_21-3-1	0.9510	0.9518	0.9992	0.9749
physics_07_21-3-2	0.9643	0.9651	0.9992	0.9818
physics_07_22-1-2	0.9820	0.9820	1.0000	0.9909
physics_07_22-1-3	1.0000	1.0000	1.0000	1.0000
physics_07_22-2-1	0.9987	1.0000	0.9987	0.9994
physics_07_22-2-3	0.9987	1.0000	0.9987	0.9994
physics_07_22-3-1	1.0000	1.0000	1.0000	1.0000
physics_07_22-3-2	0.9820	0.9820	1.0000	0.9909
physics_07_23-1-2	0.9761	0.9761	1.0000	0.9879
physics_07_23-1-3	0.9779	0.9779	1.0000	0.9888
physics_07_23-2-1	0.9963	1.0000	0.9963	0.9982
physics_07_23-2-3	0.9816	0.9815	1.0000	0.9907
physics_07_23-3-1	0.9963	1.0000	0.9963	0.9982
physics_07_23-3-2	0.9798	0.9797	1.0000	0.9897
physics_07_24-1-2	0.8208	0.8210	0.9989	0.9013
physics_07_24-1-3	0.8664	0.8668	0.9990	0.9282
physics_07_24-2-1	0.8770	0.8965	0.9750	0.9341
physics_07_24-2-3	0.8503	0.8677	0.9755	0.9185
physics_07_24-3-1	0.8957	0.8960	0.9993	0.9448
physics_07_24-3-2	0.8206	0.8208	0.9991	0.9012
physics_07_28-1-2	0.6600	0.6576	0.9995	0.7933
physics_07_28-1-3	0.6917	0.6896	0.9996	0.8161
physics_07_28-2-1	0.7311	0.9477	0.7451	0.8343
physics_07_28-2-3	0.7177	0.7816	0.8154	0.7981
physics_07_28-3-1	0.7876	0.9416	0.8168	0.8748
physics_07_28-3-2	0.7036	0.7262	0.8764	0.7943
physics_07_29-1-2	0.9093	0.9093	1.0000	0.9525
physics_07_29-1-3	0.8387	0.8387	1.0000	0.9123
physics_07_29-2-1	0.9777	0.9874	0.9901	0.9887
physics_07_29-2-3	0.8462	0.8459	0.9985	0.9159

Continued…

File	Accuracy	Precision	Recall	F-score
physics_07_29-3-1	0.9416	0.9884	0.9521	0.9699
physics_07_29-3-2	0.9023	0.9268	0.9692	0.9475
physics_07_31-1-2	0.9724	0.9740	0.9980	0.9858
physics_07_31-1-3	0.9369	0.9380	0.9979	0.9670
physics_07_31-2-1	0.9822	0.9859	0.9959	0.9909
physics_07_31-2-3	0.9408	0.9418	0.9979	0.9690
physics_07_31-3-1	0.9803	0.9899	0.9899	0.9899
physics_07_31-3-2	0.9783	0.9838	0.9939	0.9888
Min	0.6600	0.6576	0.7451	0.7933
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.9259	0.9371	0.9833	0.9577
Std Dev	0.0864	0.0775	0.0481	0.0544

F.5 #python, SAME ANNOTATOR, DIFFERENT SESSION

File	Accuracy	Precision	Recall	F-score
python_07_17-1-18-1	0.7304	0.7671	0.8481	0.8056
python_07_17-1-19-1	0.7043	0.7574	0.8124	0.7839
python_07_17-1-21-1	0.6988	0.7631	0.8195	0.7903
python_07_17-1-22-1	0.7014	0.6421	0.8526	0.7325
python_07_17-1-23-1	0.7436	0.7394	0.8711	0.7999
python_07_17-1-24-1	0.6566	0.5902	0.8736	0.7045
python_07_17-1-28-1	0.7150	0.7975	0.7676	0.7823
python_07_17-1-29-1	0.7525	0.7498	0.8722	0.8064
python_07_17-1-31-1	0.7051	0.7253	0.8467	0.7813
python_07_17-2-18-2	0.7324	0.7316	0.9087	0.8106
python_07_17-2-19-2	0.6796	0.6855	0.8738	0.7683
python_07_17-2-21-2	0.7194	0.7375	0.8958	0.8090
python_07_17-2-22-2	0.6832	0.6255	0.8886	0.7342

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_17-2-23-2	0.7282	0.7503	0.8742	0.8076
python_07_17-2-24-2	0.6384	0.5595	0.9332	0.6996
python_07_17-2-28-2	0.7169	0.6725	0.8955	0.7682
python_07_17-2-29-2	0.7292	0.7174	0.8977	0.7975
python_07_17-2-31-2	0.7039	0.7077	0.8841	0.7861
python_07_17-3-18-3	0.7385	0.7577	0.8755	0.8123
python_07_17-3-19-3	0.6968	0.6802	0.8745	0.7653
python_07_17-3-21-3	0.7273	0.7162	0.9070	0.8004
python_07_17-3-22-3	0.7119	0.6594	0.8629	0.7476
python_07_17-3-23-3	0.7352	0.7215	0.8857	0.7952
python_07_17-3-24-3	0.6573	0.5907	0.8916	0.7106
python_07_17-3-28-3	0.7453	0.7026	0.8853	0.7834
python_07_17-3-29-3	0.7386	0.7298	0.8756	0.7961
python_07_17-3-31-3	0.7029	0.6693	0.9033	0.7689
python_07_18-1-17-1	0.7261	0.7875	0.8048	0.7961
python_07_18-1-19-1	0.7030	0.7765	0.7725	0.7745
python_07_18-1-21-1	0.7022	0.7773	0.7989	0.7880
python_07_18-1-22-1	0.7555	0.7121	0.8229	0.7635
python_07_18-1-23-1	0.7709	0.7759	0.8583	0.8150
python_07_18-1-24-1	0.6882	0.6200	0.8639	0.7219
python_07_18-1-28-1	0.7326	0.8321	0.7507	0.7893
python_07_18-1-29-1	0.7897	0.8021	0.8553	0.8278
python_07_18-1-31-1	0.7446	0.7654	0.8499	0.8055
python_07_18-2-17-2	0.7535	0.8491	0.7863	0.8165
python_07_18-2-19-2	0.6996	0.7470	0.7648	0.7558
python_07_18-2-21-2	0.7471	0.8038	0.8185	0.8111
python_07_18-2-22-2	0.7680	0.7519	0.7891	0.7701
python_07_18-2-23-2	0.7522	0.8289	0.7814	0.8045
python_07_18-2-24-2	0.7467	0.6648	0.8843	0.7590
python_07_18-2-28-2	0.7904	0.7797	0.8360	0.8069
python_07_18-2-29-2	0.7827	0.8179	0.8159	0.8169
python_07_18-2-31-2	0.7523	0.7777	0.8366	0.8061

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_18-3-17-3	0.7261	0.7908	0.7937	0.7922
python_07_18-3-19-3	0.7133	0.7127	0.8252	0.7648
python_07_18-3-21-3	0.7521	0.7520	0.8785	0.8104
python_07_18-3-22-3	0.7584	0.7269	0.8193	0.7703
python_07_18-3-23-3	0.7705	0.7706	0.8609	0.8132
python_07_18-3-24-3	0.7025	0.6378	0.8548	0.7306
python_07_18-3-28-3	0.7821	0.7553	0.8598	0.8042
python_07_18-3-29-3	0.7843	0.7932	0.8519	0.8215
python_07_18-3-31-3	0.7319	0.7038	0.8804	0.7823
python_07_19-1-17-1	0.7181	0.7745	0.8120	0.7928
python_07_19-1-18-1	0.7452	0.7750	0.8640	0.8171
python_07_19-1-21-1	0.7094	0.7674	0.8328	0.7988
python_07_19-1-22-1	0.7210	0.6590	0.8666	0.7487
python_07_19-1-23-1	0.7474	0.7408	0.8775	0.8034
python_07_19-1-24-1	0.6740	0.6046	0.8785	0.7163
python_07_19-1-28-1	0.7406	0.8202	0.7826	0.8010
python_07_19-1-29-1	0.7648	0.7654	0.8681	0.8135
python_07_19-1-31-1	0.7190	0.7390	0.8477	0.7896
python_07_19-2-17-2	0.7502	0.8478	0.7822	0.8137
python_07_19-2-18-2	0.7659	0.7990	0.8398	0.8189
python_07_19-2-21-2	0.7385	0.7947	0.8169	0.8056
python_07_19-2-22-2	0.7737	0.7518	0.8067	0.7783
python_07_19-2-23-2	0.7490	0.8287	0.7755	0.8012
python_07_19-2-24-2	0.7442	0.6625	0.8827	0.7569
python_07_19-2-28-2	0.7835	0.7749	0.8267	0.8000
python_07_19-2-29-2	0.7710	0.8100	0.8028	0.8064
python_07_19-2-31-2	0.7496	0.7800	0.8262	0.8024
python_07_19-3-17-3	0.7285	0.8286	0.7405	0.7821
python_07_19-3-18-3	0.7465	0.8267	0.7691	0.7968
python_07_19-3-21-3	0.7753	0.8035	0.8302	0.8166
python_07_19-3-22-3	0.7824	0.7901	0.7625	0.7761
python_07_19-3-23-3	0.7819	0.8289	0.7866	0.8072

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_19-3-24-3	0.7390	0.6948	0.7965	0.7422
python_07_19-3-28-3	0.7979	0.8128	0.7949	0.8037
python_07_19-3-29-3	0.7926	0.8520	0.7795	0.8141
python_07_19-3-31-3	0.7416	0.7479	0.7960	0.7712
python_07_21-1-17-1	0.7105	0.7475	0.8519	0.7963
python_07_21-1-18-1	0.7237	0.7448	0.8830	0.8080
python_07_21-1-19-1	0.7041	0.7367	0.8588	0.7930
python_07_21-1-22-1	0.6948	0.6284	0.8902	0.7367
python_07_21-1-23-1	0.7189	0.7036	0.9019	0.7905
python_07_21-1-24-1	0.6480	0.5790	0.9110	0.7080
python_07_21-1-28-1	0.7382	0.7956	0.8177	0.8065
python_07_21-1-29-1	0.7468	0.7374	0.8876	0.8056
python_07_21-1-31-1	0.7118	0.7206	0.8765	0.7910
python_07_21-2-17-2	0.7474	0.8406	0.7870	0.8129
python_07_21-2-18-2	0.7565	0.7859	0.8433	0.8136
python_07_21-2-19-2	0.6943	0.7341	0.7795	0.7561
python_07_21-2-22-2	0.7578	0.7337	0.7976	0.7643
python_07_21-2-23-2	0.7418	0.8114	0.7872	0.7991
python_07_21-2-24-2	0.7370	0.6505	0.9014	0.7557
python_07_21-2-28-2	0.7793	0.7648	0.8354	0.7986
python_07_21-2-29-2	0.7716	0.8042	0.8136	0.8089
python_07_21-2-31-2	0.7387	0.7683	0.8240	0.7952
python_07_21-3-17-3	0.7247	0.8148	0.7527	0.7825
python_07_21-3-18-3	0.7490	0.8141	0.7928	0.8033
python_07_21-3-19-3	0.7172	0.7383	0.7737	0.7556
python_07_21-3-22-3	0.7828	0.7851	0.7720	0.7785
python_07_21-3-23-3	0.7764	0.8085	0.8058	0.8071
python_07_21-3-24-3	0.7377	0.6871	0.8154	0.7458
python_07_21-3-28-3	0.8110	0.8189	0.8177	0.8183
python_07_21-3-29-3	0.7884	0.8323	0.7977	0.8146
python_07_21-3-31-3	0.7483	0.7416	0.8287	0.7827
python_07_22-1-17-1	0.7219	0.8517	0.7037	0.7707

Continued…

File	Accuracy	Precision	Recall	F-score
python_07_22-1-18-1	0.7198	0.8683	0.6773	0.7610
python_07_22-1-19-1	0.6800	0.8584	0.6171	0.7180
python_07_22-1-21-1	0.6901	0.8370	0.6861	0.7541
python_07_22-1-23-1	0.7706	0.8689	0.7184	0.7865
python_07_22-1-24-1	0.7644	0.7423	0.7614	0.7517
python_07_22-1-28-1	0.6947	0.8846	0.6237	0.7316
python_07_22-1-29-1	0.7897	0.8951	0.7297	0.8040
python_07_22-1-31-1	0.7182	0.8228	0.6971	0.7547
python_07_22-2-17-2	0.7029	0.9013	0.6445	0.7516
python_07_22-2-18-2	0.7244	0.8889	0.6431	0.7463
python_07_22-2-19-2	0.6876	0.8505	0.5897	0.6965
python_07_22-2-21-2	0.7178	0.8604	0.6859	0.7633
python_07_22-2-23-2	0.7142	0.9120	0.6219	0.7395
python_07_22-2-24-2	0.7894	0.7715	0.7577	0.7645
python_07_22-2-28-2	0.7782	0.8464	0.7043	0.7689
python_07_22-2-29-2	0.7517	0.8963	0.6582	0.7590
python_07_22-2-31-2	0.7160	0.8498	0.6541	0.7392
python_07_22-3-17-3	0.7129	0.8322	0.7060	0.7639
python_07_22-3-18-3	0.7344	0.8461	0.7203	0.7781
python_07_22-3-19-3	0.7254	0.7816	0.7132	0.7459
python_07_22-3-21-3	0.7731	0.8229	0.7947	0.8085
python_07_22-3-23-3	0.7834	0.8582	0.7511	0.8010
python_07_22-3-24-3	0.7477	0.7190	0.7638	0.7407
python_07_22-3-28-3	0.7929	0.8263	0.7624	0.7931
python_07_22-3-29-3	0.7901	0.8765	0.7447	0.8053
python_07_22-3-31-3	0.7376	0.7623	0.7562	0.7592
python_07_23-1-17-1	0.7233	0.8237	0.7422	0.7808
python_07_23-1-18-1	0.7430	0.8253	0.7736	0.7986
python_07_23-1-19-1	0.7109	0.8217	0.7179	0.7663
python_07_23-1-21-1	0.6929	0.8036	0.7366	0.7686
python_07_23-1-22-1	0.7796	0.7693	0.7721	0.7707
python_07_23-1-24-1	0.7411	0.6886	0.8164	0.7471

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_23-1-28-1	0.7190	0.8694	0.6810	0.7638
python_07_23-1-29-1	0.7922	0.8462	0.7925	0.8185
python_07_23-1-31-1	0.7341	0.7970	0.7681	0.7823
python_07_23-2-17-2	0.7545	0.8259	0.8209	0.8234
python_07_23-2-18-2	0.7620	0.7752	0.8765	0.8227
python_07_23-2-19-2	0.6972	0.7190	0.8238	0.7678
python_07_23-2-21-2	0.7301	0.7678	0.8501	0.8069
python_07_23-2-22-2	0.7496	0.7082	0.8358	0.7667
python_07_23-2-24-2	0.7101	0.6207	0.9184	0.7408
python_07_23-2-28-2	0.7704	0.7414	0.8624	0.7974
python_07_23-2-29-2	0.7757	0.7857	0.8560	0.8193
python_07_23-2-31-2	0.7451	0.7586	0.8591	0.8058
python_07_23-3-17-3	0.7190	0.8160	0.7398	0.7760
python_07_23-3-18-3	0.7405	0.8152	0.7742	0.7941
python_07_23-3-19-3	0.7242	0.7547	0.7584	0.7565
python_07_23-3-21-3	0.7710	0.7943	0.8368	0.8150
python_07_23-3-22-3	0.7841	0.7893	0.7684	0.7787
python_07_23-3-24-3	0.7355	0.6862	0.8095	0.7428
python_07_23-3-28-3	0.7934	0.8035	0.7982	0.8009
python_07_23-3-29-3	0.7869	0.8361	0.7889	0.8118
python_07_23-3-31-3	0.7444	0.7466	0.8067	0.7755
python_07_24-1-17-1	0.6944	0.8883	0.6175	0.7286
python_07_24-1-18-1	0.6813	0.9175	0.5672	0.7010
python_07_24-1-19-1	0.6425	0.9137	0.5064	0.6516
python_07_24-1-21-1	0.6677	0.8772	0.6050	0.7161
python_07_24-1-22-1	0.7958	0.8925	0.6527	0.7540
python_07_24-1-23-1	0.7576	0.9284	0.6371	0.7556
python_07_24-1-28-1	0.6531	0.9087	0.5335	0.6723
python_07_24-1-29-1	0.7670	0.9433	0.6446	0.7658
python_07_24-1-31-1	0.6810	0.8612	0.5808	0.6938
python_07_24-2-17-2	0.6708	0.8986	0.5950	0.7159
python_07_24-2-18-2	0.7201	0.9085	0.6181	0.7356

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_24-2-19-2	0.6746	0.8650	0.5505	0.6728
python_07_24-2-21-2	0.7233	0.8984	0.6573	0.7591
python_07_24-2-22-2	0.7738	0.8770	0.6288	0.7324
python_07_24-2-23-2	0.6975	0.9192	0.5879	0.7172
python_07_24-2-28-2	0.7835	0.8973	0.6625	0.7622
python_07_24-2-29-2	0.7430	0.9242	0.6180	0.7407
python_07_24-2-31-2	0.7136	0.8693	0.6293	0.7301
python_07_24-3-17-3	0.7072	0.8799	0.6427	0.7428
python_07_24-3-18-3	0.7173	0.8945	0.6378	0.7447
python_07_24-3-19-3	0.7156	0.8400	0.6136	0.7092
python_07_24-3-21-3	0.7835	0.8807	0.7413	0.8050
python_07_24-3-22-3	0.7885	0.8756	0.6670	0.7572
python_07_24-3-23-3	0.7693	0.9042	0.6740	0.7723
python_07_24-3-28-3	0.7891	0.8766	0.6923	0.7736
python_07_24-3-29-3	0.7570	0.8987	0.6570	0.7591
python_07_24-3-31-3	0.7376	0.8144	0.6739	0.7375
python_07_28-1-17-1	0.7053	0.7368	0.8654	0.7959
python_07_28-1-18-1	0.7318	0.7384	0.9179	0.8185
python_07_28-1-19-1	0.7155	0.7274	0.9104	0.8086
python_07_28-1-21-1	0.7146	0.7430	0.8989	0.8135
python_07_28-1-22-1	0.6733	0.6056	0.9137	0.7284
python_07_28-1-23-1	0.7190	0.6948	0.9312	0.7959
python_07_28-1-24-1	0.6109	0.5505	0.9233	0.6898
python_07_28-1-29-1	0.7379	0.7150	0.9254	0.8067
python_07_28-1-31-1	0.7049	0.7030	0.9101	0.7933
python_07_28-2-17-2	0.7465	0.8548	0.7666	0.8083
python_07_28-2-18-2	0.7635	0.8001	0.8328	0.8161
python_07_28-2-19-2	0.6960	0.7457	0.7585	0.7521
python_07_28-2-21-2	0.7544	0.8158	0.8134	0.8146
python_07_28-2-22-2	0.7621	0.7533	0.7685	0.7608
python_07_28-2-23-2	0.7424	0.8296	0.7614	0.7940
python_07_28-2-24-2	0.7596	0.6818	0.8761	0.7668

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_28-2-29-2	0.7822	0.8242	0.8050	0.8145
python_07_28-2-31-2	0.7464	0.7834	0.8126	0.7978
python_07_28-3-17-3	0.7247	0.8090	0.7613	0.7844
python_07_28-3-18-3	0.7487	0.8063	0.8044	0.8054
python_07_28-3-19-3	0.7102	0.7273	0.7796	0.7525
python_07_28-3-21-3	0.7780	0.7899	0.8607	0.8238
python_07_28-3-22-3	0.7748	0.7727	0.7715	0.7721
python_07_28-3-23-3	0.7775	0.8041	0.8154	0.8097
python_07_28-3-24-3	0.7279	0.6743	0.8188	0.7396
python_07_28-3-29-3	0.7863	0.8220	0.8084	0.8151
python_07_28-3-31-3	0.7498	0.7372	0.8434	0.7867
python_07_29-1-17-1	0.7185	0.8035	0.7628	0.7826
python_07_29-1-18-1	0.7459	0.8126	0.7983	0.8054
python_07_29-1-19-1	0.7034	0.7943	0.7432	0.7679
python_07_29-1-21-1	0.7017	0.7940	0.7689	0.7812
python_07_29-1-22-1	0.7624	0.7351	0.7888	0.7610
python_07_29-1-23-1	0.7757	0.7986	0.8272	0.8126
python_07_29-1-24-1	0.7228	0.6602	0.8414	0.7399
python_07_29-1-28-1	0.7257	0.8523	0.7123	0.7760
python_07_29-1-31-1	0.7374	0.7758	0.8128	0.7939
python_07_29-2-17-2	0.7474	0.8421	0.7849	0.8125
python_07_29-2-18-2	0.7621	0.7904	0.8472	0.8178
python_07_29-2-19-2	0.6956	0.7355	0.7797	0.7569
python_07_29-2-21-2	0.7413	0.7920	0.8273	0.8092
python_07_29-2-22-2	0.7611	0.7369	0.8007	0.7675
python_07_29-2-23-2	0.7470	0.8180	0.7874	0.8024
python_07_29-2-24-2	0.7412	0.6548	0.9017	0.7587
python_07_29-2-28-2	0.7900	0.7730	0.8481	0.8088
python_07_29-2-31-2	0.7473	0.7688	0.8428	0.8041
python_07_29-3-17-3	0.7133	0.7931	0.7635	0.7780
python_07_29-3-18-3	0.7396	0.7939	0.8066	0.8002
python_07_29-3-19-3	0.7134	0.7220	0.8014	0.7596

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_29-3-21-3	0.7563	0.7616	0.8671	0.8109
python_07_29-3-22-3	0.7625	0.7425	0.7956	0.7681
python_07_29-3-23-3	0.7652	0.7810	0.8274	0.8036
python_07_29-3-24-3	0.7076	0.6461	0.8407	0.7307
python_07_29-3-28-3	0.7871	0.7686	0.8454	0.8052
python_07_29-3-31-3	0.7285	0.7128	0.8437	0.7728
python_07_31-1-17-1	0.7228	0.7801	0.8113	0.7954
python_07_31-1-18-1	0.7537	0.7965	0.8408	0.8181
python_07_31-1-19-1	0.7046	0.7800	0.7697	0.7748
python_07_31-1-21-1	0.7028	0.7811	0.7932	0.7871
python_07_31-1-22-1	0.7544	0.7107	0.8227	0.7626
python_07_31-1-23-1	0.7710	0.7768	0.8568	0.8148
python_07_31-1-24-1	0.6896	0.6240	0.8489	0.7193
python_07_31-1-28-1	0.7300	0.8288	0.7502	0.7875
python_07_31-1-29-1	0.7790	0.7979	0.8383	0.8176
python_07_31-2-17-2	0.7431	0.8327	0.7904	0.8110
python_07_31-2-18-2	0.7637	0.7997	0.8338	0.8164
python_07_31-2-19-2	0.6974	0.7498	0.7535	0.7516
python_07_31-2-21-2	0.7350	0.7965	0.8065	0.8015
python_07_31-2-22-2	0.7574	0.7400	0.7822	0.7605
python_07_31-2-23-2	0.7481	0.8252	0.7787	0.8013
python_07_31-2-24-2	0.7300	0.6495	0.8722	0.7446
python_07_31-2-28-2	0.7662	0.7517	0.8265	0.7873
python_07_31-2-29-2	0.7716	0.8094	0.8050	0.8072
python_07_31-3-17-3	0.7138	0.8259	0.7160	0.7670
python_07_31-3-18-3	0.7452	0.8320	0.7592	0.7939
python_07_31-3-19-3	0.7189	0.7604	0.7337	0.7468
python_07_31-3-21-3	0.7849	0.8290	0.8104	0.8196
python_07_31-3-22-3	0.7756	0.8059	0.7194	0.7602
python_07_31-3-23-3	0.7807	0.8297	0.7830	0.8057
python_07_31-3-24-3	0.7444	0.7120	0.7694	0.7396
python_07_31-3-28-3	0.7999	0.8308	0.7728	0.8008

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_31-3-29-3	0.7665	0.8395	0.7408	0.7871
Min	0.6109	0.5505	0.5064	0.6516
Max	0.8110	0.9433	0.9332	0.8278
Avg	0.7377	0.7794	0.7911	0.7780
Std Dev	0.0343	0.0742	0.0835	0.0331

F.6 #python, SAME SESSION, DIFFERENT ANNOTATOR

File	Accuracy	Precision	Recall	F-score
python_07_17-1-2	0.7715	0.8337	0.8399	0.8368
python_07_17-1-3	0.7408	0.7838	0.8368	0.8095
python_07_17-2-1	0.7308	0.7601	0.8689	0.8109
python_07_17-2-3	0.7304	0.7558	0.8720	0.8097
python_07_17-3-1	0.7323	0.7782	0.8348	0.8055
python_07_17-3-2	0.7644	0.8240	0.8419	0.8329
python_07_18-1-2	0.7637	0.7828	0.8651	0.8219
python_07_18-1-3	0.7477	0.7830	0.8436	0.8122
python_07_18-2-1	0.7534	0.8129	0.8125	0.8127
python_07_18-2-3	0.7534	0.8037	0.8184	0.8110
python_07_18-3-1	0.7548	0.8012	0.8348	0.8177
python_07_18-3-2	0.7679	0.7900	0.8604	0.8237
python_07_19-1-2	0.7083	0.7189	0.8539	0.7806
python_07_19-1-3	0.7045	0.6866	0.8773	0.7704
python_07_19-2-1	0.7259	0.8089	0.7657	0.7867
python_07_19-2-3	0.7277	0.7342	0.8122	0.7712
python_07_19-3-1	0.7146	0.8356	0.7068	0.7658
python_07_19-3-2	0.7160	0.7899	0.7258	0.7565
python_07_21-1-2	0.7365	0.7509	0.9021	0.8196
python_07_21-1-3	0.7226	0.7042	0.9309	0.8018

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_21-2-1	0.7063	0.7875	0.7888	0.7882
python_07_21-2-3	0.7618	0.7628	0.8778	0.8163
python_07_21-3-1	0.7057	0.8109	0.7501	0.7793
python_07_21-3-2	0.7555	0.8269	0.7986	0.8125
python_07_22-1-2	0.7897	0.8316	0.7182	0.7708
python_07_22-1-3	0.7909	0.8355	0.7186	0.7727
python_07_22-2-1	0.7926	0.8425	0.6980	0.7635
python_07_22-2-3	0.7912	0.8594	0.6906	0.7658
python_07_22-3-1	0.7870	0.7986	0.7433	0.7699
python_07_22-3-2	0.7899	0.8163	0.7399	0.7762
python_07_23-1-2	0.7470	0.8529	0.7397	0.7923
python_07_23-1-3	0.7869	0.8247	0.8038	0.8141
python_07_23-2-1	0.7670	0.7682	0.8647	0.8136
python_07_23-2-3	0.7676	0.7629	0.8701	0.8130
python_07_23-3-1	0.7860	0.8330	0.7957	0.8139
python_07_23-3-2	0.7496	0.8577	0.7387	0.7938
python_07_24-1-2	0.8015	0.8245	0.7115	0.7638
python_07_24-1-3	0.7647	0.8038	0.6632	0.7268
python_07_24-2-1	0.7847	0.8262	0.6844	0.7486
python_07_24-2-3	0.7649	0.8050	0.6621	0.7266
python_07_24-3-1	0.7838	0.8013	0.7162	0.7563
python_07_24-3-2	0.8026	0.8029	0.7452	0.7730
python_07_28-1-2	0.7301	0.6771	0.9267	0.7825
python_07_28-1-3	0.7325	0.6764	0.9316	0.7838
python_07_28-2-1	0.7288	0.8694	0.6984	0.7746
python_07_28-2-3	0.8101	0.8085	0.8324	0.8203
python_07_28-3-1	0.7229	0.8718	0.6854	0.7675
python_07_28-3-2	0.8066	0.8149	0.8161	0.8155
python_07_29-1-2	0.7875	0.8241	0.8165	0.8203
python_07_29-1-3	0.7988	0.8241	0.8324	0.8282
python_07_29-2-1	0.7998	0.8230	0.8425	0.8326
python_07_29-2-3	0.7960	0.8129	0.8441	0.8282

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_29-3-1	0.7933	0.8207	0.8322	0.8264
python_07_29-3-2	0.7812	0.8131	0.8203	0.8167
python_07_31-1-2	0.7540	0.7680	0.8599	0.8114
python_07_31-1-3	0.7372	0.7063	0.8896	0.7874
python_07_31-2-1	0.7508	0.7800	0.8348	0.8065
python_07_31-2-3	0.7428	0.7177	0.8734	0.7879
python_07_31-3-1	0.7344	0.8118	0.7460	0.7775
python_07_31-3-2	0.7461	0.8162	0.7582	0.7861
Min	0.7045	0.6764	0.6621	0.7266
Max	0.8101	0.8718	0.9316	0.8368
Avg	0.7583	0.7952	0.8011	0.7944
Std Dev	0.0297	0.0459	0.0717	0.0264

APPENDIX G: MAXIMUM ENTROPY WITH LDA CLASSIFICATION RESULTS

The following are the full classification results using the maximum entropy model with LDA augmentation. The results format is the same as in Appendix F.

G.1 ##iphone, SAME ANNOTATOR, DIFFERENT SESSION

File	Accuracy	Precision	Recall	F-score
iphone_07_17-1-18-1	0.8915	0.8915	1.0000	0.9426
iphone_07_17-1-19-1	0.9126	0.9126	1.0000	0.9543
iphone_07_17-1-21-1	0.8300	0.8300	1.0000	0.9071
iphone_07_17-1-22-1	0.8055	0.8055	1.0000	0.8923
iphone_07_17-1-23-1	0.7786	0.7786	1.0000	0.8755
iphone_07_17-1-24-1	0.7255	0.7255	1.0000	0.8409
iphone_07_17-1-28-1	0.9578	0.9578	1.0000	0.9784
iphone_07_17-1-29-1	0.9366	0.9366	1.0000	0.9673
iphone_07_17-1-31-1	0.8056	0.8056	1.0000	0.8924
iphone_07_17-2-18-2	0.9500	0.9501	0.9998	0.9743
iphone_07_17-2-19-2	0.9708	0.9715	0.9992	0.9852
iphone_07_17-2-21-2	0.9082	0.9088	0.9992	0.9519
iphone_07_17-2-22-2	0.8446	0.8459	0.9981	0.9157
iphone_07_17-2-23-2	0.6941	0.6941	1.0000	0.8194
iphone_07_17-2-24-2	0.7190	0.7192	0.9993	0.8364
iphone_07_17-2-28-2	0.9413	0.9413	1.0000	0.9698
iphone_07_17-2-29-2	0.9259	0.9259	1.0000	0.9615
iphone_07_17-2-31-2	0.8106	0.8133	0.9959	0.8954
iphone_07_17-3-18-3	0.8028	0.8028	1.0000	0.8906
iphone_07_17-3-19-3	0.8402	0.8402	1.0000	0.9132
iphone_07_17-3-21-3	0.8483	0.8483	1.0000	0.9179
iphone_07_17-3-22-3	0.7619	0.7619	1.0000	0.8648

Continued...

File	Accuracy	Precision	Recall	F-score
iphone_07_17-3-23-3	0.7097	0.7097	1.0000	0.8302
iphone_07_17-3-24-3	0.7436	0.7436	1.0000	0.8530
iphone_07_17-3-28-3	0.9635	0.9635	1.0000	0.9814
iphone_07_17-3-29-3	0.8728	0.8728	1.0000	0.9321
iphone_07_17-3-31-3	0.7940	0.7940	1.0000	0.8852
iphone_07_18-1-17-1	0.9854	0.9854	1.0000	0.9927
iphone_07_18-1-19-1	0.9000	0.9144	0.9823	0.9471
iphone_07_18-1-21-1	0.8293	0.8350	0.9900	0.9059
iphone_07_18-1-22-1	0.8039	0.8119	0.9846	0.8900
iphone_07_18-1-23-1	0.7828	0.7823	0.9991	0.8775
iphone_07_18-1-24-1	0.7296	0.7317	0.9904	0.8416
iphone_07_18-1-28-1	0.9578	0.9598	0.9978	0.9784
iphone_07_18-1-29-1	0.9351	0.9420	0.9918	0.9663
iphone_07_18-1-31-1	0.8056	0.8056	1.0000	0.8924
iphone_07_18-2-17-2	0.9549	0.9549	1.0000	0.9769
iphone_07_18-2-19-2	0.9712	0.9715	0.9997	0.9854
iphone_07_18-2-21-2	0.9082	0.9082	1.0000	0.9519
iphone_07_18-2-22-2	0.8430	0.8458	0.9960	0.9148
iphone_07_18-2-23-2	0.6927	0.6937	0.9980	0.8184
iphone_07_18-2-24-2	0.7217	0.7211	0.9997	0.8378
iphone_07_18-2-28-2	0.9406	0.9406	1.0000	0.9694
iphone_07_18-2-29-2	0.9259	0.9259	1.0000	0.9615
iphone_07_18-2-31-2	0.8140	0.8140	1.0000	0.8974
iphone_07_18-3-17-3	0.9563	0.9618	0.9939	0.9776
iphone_07_18-3-19-3	0.8392	0.8641	0.9595	0.9093
iphone_07_18-3-21-3	0.8369	0.8653	0.9567	0.9087
iphone_07_18-3-22-3	0.7657	0.7848	0.9541	0.8612
iphone_07_18-3-23-3	0.7289	0.7242	0.9980	0.8394
iphone_07_18-3-24-3	0.7487	0.7627	0.9612	0.8505
iphone_07_18-3-28-3	0.9628	0.9682	0.9941	0.9810
iphone_07_18-3-29-3	0.8723	0.8915	0.9719	0.9300
iphone_07_18-3-31-3	0.8023	0.8017	0.9979	0.8891

Continued. . .

File	Accuracy	Precision	Recall	F-score
iphone_07_19-1-17-1	0.9854	0.9854	1.0000	0.9927
iphone_07_19-1-18-1	0.8918	0.8921	0.9995	0.9427
iphone_07_19-1-21-1	0.8300	0.8305	0.9991	0.9071
iphone_07_19-1-22-1	0.8064	0.8066	0.9993	0.8927
iphone_07_19-1-23-1	0.7800	0.7797	1.0000	0.8762
iphone_07_19-1-24-1	0.7284	0.7281	0.9986	0.8422
iphone_07_19-1-28-1	0.9557	0.9577	0.9978	0.9773
iphone_07_19-1-29-1	0.9366	0.9366	1.0000	0.9673
iphone_07_19-1-31-1	0.8056	0.8056	1.0000	0.8924
iphone_07_19-2-17-2	0.9549	0.9549	1.0000	0.9769
iphone_07_19-2-18-2	0.9501	0.9501	1.0000	0.9744
iphone_07_19-2-21-2	0.9084	0.9086	0.9997	0.9520
iphone_07_19-2-22-2	0.8464	0.8465	0.9999	0.9168
iphone_07_19-2-23-2	0.6941	0.6944	0.9990	0.8193
iphone_07_19-2-24-2	0.7194	0.7195	0.9993	0.8366
iphone_07_19-2-28-2	0.9399	0.9406	0.9992	0.9690
iphone_07_19-2-29-2	0.9259	0.9259	1.0000	0.9615
iphone_07_19-2-31-2	0.8140	0.8140	1.0000	0.8974
iphone_07_19-3-17-3	0.9578	0.9592	0.9985	0.9784
iphone_07_19-3-18-3	0.8114	0.8147	0.9904	0.8940
iphone_07_19-3-21-3	0.8439	0.8549	0.9828	0.9144
iphone_07_19-3-22-3	0.7697	0.7741	0.9853	0.8670
iphone_07_19-3-23-3	0.7126	0.7139	0.9930	0.8306
iphone_07_19-3-24-3	0.7494	0.7547	0.9823	0.8536
iphone_07_19-3-28-3	0.9628	0.9662	0.9963	0.9810
iphone_07_19-3-29-3	0.8769	0.8814	0.9924	0.9336
iphone_07_19-3-31-3	0.8023	0.8007	1.0000	0.8893
iphone_07_21-1-17-1	0.9767	0.9853	0.9911	0.9882
iphone_07_21-1-18-1	0.8796	0.9076	0.9629	0.9345
iphone_07_21-1-19-1	0.8804	0.9160	0.9567	0.9359
iphone_07_21-1-22-1	0.7956	0.8202	0.9559	0.8828
iphone_07_21-1-23-1	0.7757	0.7803	0.9909	0.8731

Continued...

File	Accuracy	Precision	Recall	F-score
iphone_07_21-1-24-1	0.7264	0.7362	0.9706	0.8373
iphone_07_21-1-28-1	0.9521	0.9602	0.9910	0.9754
iphone_07_21-1-29-1	0.9157	0.9455	0.9656	0.9555
iphone_07_21-1-31-1	0.7857	0.8017	0.9753	0.8800
iphone_07_21-2-17-2	0.9534	0.9548	0.9985	0.9762
iphone_07_21-2-18-2	0.9444	0.9519	0.9916	0.9713
iphone_07_21-2-19-2	0.9621	0.9748	0.9865	0.9806
iphone_07_21-2-22-2	0.8408	0.8489	0.9876	0.9130
iphone_07_21-2-23-2	0.6913	0.6932	0.9959	0.8175
iphone_07_21-2-24-2	0.7230	0.7262	0.9869	0.8367
iphone_07_21-2-28-2	0.9406	0.9432	0.9970	0.9693
iphone_07_21-2-29-2	0.9264	0.9299	0.9956	0.9616
iphone_07_21-2-31-2	0.8289	0.8263	1.0000	0.9049
iphone_07_21-3-17-3	0.9549	0.9618	0.9924	0.9768
iphone_07_21-3-18-3	0.8081	0.8160	0.9824	0.8915
iphone_07_21-3-19-3	0.8359	0.8489	0.9789	0.9093
iphone_07_21-3-22-3	0.7649	0.7734	0.9780	0.8637
iphone_07_21-3-23-3	0.7069	0.7101	0.9920	0.8277
iphone_07_21-3-24-3	0.7459	0.7531	0.9793	0.8515
iphone_07_21-3-28-3	0.9578	0.9646	0.9926	0.9784
iphone_07_21-3-29-3	0.8728	0.8833	0.9842	0.9310
iphone_07_21-3-31-3	0.8023	0.8027	0.9958	0.8889
iphone_07_22-1-17-1	0.9854	0.9854	1.0000	0.9927
iphone_07_22-1-18-1	0.8913	0.8965	0.9926	0.9421
iphone_07_22-1-19-1	0.9038	0.9166	0.9841	0.9492
iphone_07_22-1-21-1	0.8330	0.8364	0.9930	0.9080
iphone_07_22-1-23-1	0.7729	0.7817	0.9827	0.8708
iphone_07_22-1-24-1	0.7304	0.7356	0.9810	0.8408
iphone_07_22-1-28-1	0.9535	0.9596	0.9933	0.9761
iphone_07_22-1-29-1	0.9356	0.9434	0.9907	0.9665
iphone_07_22-1-31-1	0.8056	0.8056	1.0000	0.8924
iphone_07_22-2-17-2	0.9549	0.9549	1.0000	0.9769

Continued...

File	Accuracy	Precision	Recall	F-score
iphone_07_22-2-18-2	0.9426	0.9530	0.9884	0.9703
iphone_07_22-2-19-2	0.9560	0.9721	0.9829	0.9775
iphone_07_22-2-21-2	0.9067	0.9121	0.9930	0.9508
iphone_07_22-2-23-2	0.6962	0.6956	1.0000	0.8205
iphone_07_22-2-24-2	0.7218	0.7245	0.9895	0.8365
iphone_07_22-2-28-2	0.9421	0.9420	1.0000	0.9701
iphone_07_22-2-29-2	0.9239	0.9306	0.9917	0.9602
iphone_07_22-2-31-2	0.8140	0.8150	0.9980	0.8972
iphone_07_22-3-17-3	0.9563	0.9591	0.9970	0.9777
iphone_07_22-3-18-3	0.8065	0.8215	0.9696	0.8894
iphone_07_22-3-19-3	0.8293	0.8602	0.9515	0.9035
iphone_07_22-3-21-3	0.8381	0.8632	0.9615	0.9097
iphone_07_22-3-23-3	0.7140	0.7229	0.9680	0.8277
iphone_07_22-3-24-3	0.7522	0.7692	0.9526	0.8512
iphone_07_22-3-28-3	0.9506	0.9664	0.9829	0.9746
iphone_07_22-3-29-3	0.8656	0.8916	0.9631	0.9260
iphone_07_22-3-31-3	0.8023	0.8037	0.9937	0.8887
iphone_07_23-1-17-1	0.9723	0.9853	0.9867	0.9860
iphone_07_23-1-18-1	0.8615	0.9172	0.9284	0.9228
iphone_07_23-1-19-1	0.8287	0.9217	0.8877	0.9044
iphone_07_23-1-21-1	0.8069	0.8549	0.9243	0.8882
iphone_07_23-1-22-1	0.7712	0.8276	0.9043	0.8643
iphone_07_23-1-24-1	0.7363	0.7574	0.9367	0.8375
iphone_07_23-1-28-1	0.9342	0.9615	0.9701	0.9658
iphone_07_23-1-29-1	0.8682	0.9518	0.9051	0.9279
iphone_07_23-1-31-1	0.8040	0.8115	0.9856	0.8901
iphone_07_23-2-17-2	0.8617	0.9576	0.8948	0.9251
iphone_07_23-2-18-2	0.7566	0.9582	0.7778	0.8586
iphone_07_23-2-19-2	0.7473	0.9732	0.7608	0.8540
iphone_07_23-2-21-2	0.7869	0.9356	0.8220	0.8751
iphone_07_23-2-22-2	0.7568	0.8862	0.8176	0.8505
iphone_07_23-2-24-2	0.6949	0.7684	0.8242	0.7953

Continued…

File	Accuracy	Precision	Recall	F-score
iphone_07_23-2-28-2	0.8119	0.9413	0.8532	0.8951
iphone_07_23-2-29-2	0.7859	0.9462	0.8151	0.8758
iphone_07_23-2-31-2	0.8189	0.8457	0.9510	0.8953
iphone_07_23-3-17-3	0.8879	0.9575	0.9241	0.9405
iphone_07_23-3-18-3	0.7550	0.8613	0.8282	0.8444
iphone_07_23-3-19-3	0.7256	0.8848	0.7741	0.8258
iphone_07_23-3-21-3	0.7400	0.8852	0.7968	0.8387
iphone_07_23-3-22-3	0.7040	0.8190	0.7849	0.8016
iphone_07_23-3-24-3	0.6985	0.7929	0.8048	0.7988
iphone_07_23-3-28-3	0.8755	0.9703	0.8983	0.9329
iphone_07_23-3-29-3	0.7767	0.9129	0.8226	0.8654
iphone_07_23-3-31-3	0.8173	0.8370	0.9561	0.8926
iphone_07_24-1-17-1	0.9709	0.9852	0.9852	0.9852
iphone_07_24-1-18-1	0.8737	0.9127	0.9491	0.9305
iphone_07_24-1-19-1	0.8530	0.9220	0.9165	0.9192
iphone_07_24-1-21-1	0.8198	0.8556	0.9419	0.8967
iphone_07_24-1-22-1	0.7858	0.8323	0.9193	0.8736
iphone_07_24-1-23-1	0.7928	0.7940	0.9909	0.8816
iphone_07_24-1-28-1	0.9499	0.9642	0.9843	0.9741
iphone_07_24-1-29-1	0.9121	0.9556	0.9504	0.9530
iphone_07_24-1-31-1	0.7940	0.8096	0.9732	0.8839
iphone_07_24-2-17-2	0.9491	0.9559	0.9924	0.9738
iphone_07_24-2-18-2	0.9019	0.9537	0.9425	0.9481
iphone_07_24-2-19-2	0.8898	0.9742	0.9106	0.9414
iphone_07_24-2-21-2	0.8807	0.9272	0.9426	0.9349
iphone_07_24-2-22-2	0.8193	0.8693	0.9255	0.8965
iphone_07_24-2-23-2	0.6934	0.7090	0.9468	0.8109
iphone_07_24-2-28-2	0.9299	0.9478	0.9795	0.9634
iphone_07_24-2-29-2	0.9070	0.9425	0.9581	0.9502
iphone_07_24-2-31-2	0.8040	0.8207	0.9714	0.8897
iphone_07_24-3-17-3	0.9418	0.9585	0.9818	0.9700
iphone_07_24-3-18-3	0.8143	0.8246	0.9764	0.8941

Continued. . .

File	Accuracy	Precision	Recall	F-score
iphone_07_24-3-19-3	0.8296	0.8566	0.9575	0.9042
iphone_07_24-3-21-3	0.8393	0.8650	0.9604	0.9102
iphone_07_24-3-22-3	0.7713	0.7850	0.9639	0.8653
iphone_07_24-3-23-3	0.7225	0.7221	0.9900	0.8351
iphone_07_24-3-28-3	0.9542	0.9665	0.9866	0.9765
iphone_07_24-3-29-3	0.8687	0.8849	0.9766	0.9285
iphone_07_24-3-31-3	0.8056	0.8044	0.9979	0.8908
iphone_07_28-1-17-1	0.9854	0.9854	1.0000	0.9927
iphone_07_28-1-18-1	0.8908	0.8930	0.9969	0.9421
iphone_07_28-1-19-1	0.9110	0.9136	0.9968	0.9534
iphone_07_28-1-21-1	0.8318	0.8324	0.9982	0.9078
iphone_07_28-1-22-1	0.8061	0.8078	0.9963	0.8922
iphone_07_28-1-23-1	0.7793	0.7795	0.9991	0.8757
iphone_07_28-1-24-1	0.7289	0.7285	0.9983	0.8423
iphone_07_28-1-29-1	0.9356	0.9379	0.9973	0.9667
iphone_07_28-1-31-1	0.8073	0.8070	1.0000	0.8932
iphone_07_28-2-17-2	0.9491	0.9559	0.9924	0.9738
iphone_07_28-2-18-2	0.9420	0.9510	0.9899	0.9701
iphone_07_28-2-19-2	0.9632	0.9750	0.9875	0.9812
iphone_07_28-2-21-2	0.9070	0.9131	0.9920	0.9509
iphone_07_28-2-22-2	0.8441	0.8504	0.9899	0.9149
iphone_07_28-2-23-2	0.6969	0.6966	0.9980	0.8205
iphone_07_28-2-24-2	0.7172	0.7218	0.9874	0.8339
iphone_07_28-2-29-2	0.9249	0.9285	0.9956	0.9609
iphone_07_28-2-31-2	0.8156	0.8174	0.9959	0.8979
iphone_07_28-3-17-3	0.9563	0.9591	0.9970	0.9777
iphone_07_28-3-18-3	0.8070	0.8068	0.9986	0.8925
iphone_07_28-3-19-3	0.8408	0.8425	0.9969	0.9132
iphone_07_28-3-21-3	0.8449	0.8490	0.9940	0.9158
iphone_07_28-3-22-3	0.7622	0.7643	0.9947	0.8644
iphone_07_28-3-23-3	0.7126	0.7117	1.0000	0.8316
iphone_07_28-3-24-3	0.7458	0.7465	0.9965	0.8536

Continued. . .

File	Accuracy	Precision	Recall	F-score
iphone_07_28-3-29-3	0.8743	0.8760	0.9971	0.9326
iphone_07_28-3-31-3	0.7990	0.7980	1.0000	0.8877
iphone_07_29-1-17-1	0.9854	0.9854	1.0000	0.9927
iphone_07_29-1-18-1	0.8907	0.8917	0.9986	0.9421
iphone_07_29-1-19-1	0.9101	0.9130	0.9965	0.9529
iphone_07_29-1-21-1	0.8291	0.8302	0.9982	0.9065
iphone_07_29-1-22-1	0.8069	0.8073	0.9986	0.8928
iphone_07_29-1-23-1	0.7800	0.7797	1.0000	0.8762
iphone_07_29-1-24-1	0.7272	0.7271	0.9989	0.8416
iphone_07_29-1-28-1	0.9585	0.9585	1.0000	0.9788
iphone_07_29-1-31-1	0.8056	0.8056	1.0000	0.8924
iphone_07_29-2-17-2	0.9549	0.9549	1.0000	0.9769
iphone_07_29-2-18-2	0.9485	0.9501	0.9983	0.9736
iphone_07_29-2-19-2	0.9706	0.9738	0.9965	0.9850
iphone_07_29-2-21-2	0.9070	0.9099	0.9962	0.9511
iphone_07_29-2-22-2	0.8437	0.8473	0.9945	0.9150
iphone_07_29-2-23-2	0.6934	0.6944	0.9969	0.8186
iphone_07_29-2-24-2	0.7190	0.7208	0.9944	0.8358
iphone_07_29-2-28-2	0.9413	0.9419	0.9992	0.9697
iphone_07_29-2-31-2	0.8206	0.8194	1.0000	0.9007
iphone_07_29-3-17-3	0.9054	0.9569	0.9439	0.9503
iphone_07_29-3-18-3	0.7991	0.8165	0.9670	0.8854
iphone_07_29-3-19-3	0.8205	0.8575	0.9432	0.8983
iphone_07_29-3-21-3	0.8332	0.8676	0.9480	0.9060
iphone_07_29-3-22-3	0.7476	0.7761	0.9400	0.8502
iphone_07_29-3-23-3	0.6991	0.7121	0.9670	0.8202
iphone_07_29-3-24-3	0.7291	0.7594	0.9304	0.8363
iphone_07_29-3-28-3	0.9349	0.9666	0.9659	0.9662
iphone_07_29-3-31-3	0.8173	0.8251	0.9770	0.8946
iphone_07_31-1-17-1	0.7802	0.9981	0.7784	0.8747
iphone_07_31-1-18-1	0.7734	0.9025	0.8362	0.8681
iphone_07_31-1-19-1	0.8064	0.9118	0.8722	0.8916

Continued...

File	Accuracy	Precision	Recall	F-score
iphone_07_31-1-21-1	0.7904	0.8527	0.9035	0.8774
iphone_07_31-1-22-1	0.7283	0.8140	0.8589	0.8359
iphone_07_31-1-23-1	0.7268	0.7760	0.9125	0.8387
iphone_07_31-1-24-1	0.6675	0.7248	0.8733	0.7921
iphone_07_31-1-28-1	0.8712	0.9610	0.9022	0.9307
iphone_07_31-1-29-1	0.8370	0.9401	0.8822	0.9102
iphone_07_31-2-17-2	0.8239	0.9542	0.8567	0.9028
iphone_07_31-2-18-2	0.8129	0.9544	0.8434	0.8955
iphone_07_31-2-19-2	0.8452	0.9746	0.8631	0.9155
iphone_07_31-2-21-2	0.8147	0.9231	0.8684	0.8949
iphone_07_31-2-22-2	0.7484	0.8592	0.8404	0.8497
iphone_07_31-2-23-2	0.6899	0.7008	0.9652	0.8120
iphone_07_31-2-24-2	0.6717	0.7216	0.8850	0.7950
iphone_07_31-2-28-2	0.8827	0.9458	0.9285	0.9371
iphone_07_31-2-29-2	0.8176	0.9307	0.8675	0.8980
iphone_07_31-3-17-3	0.8210	0.9685	0.8407	0.9001
iphone_07_31-3-18-3	0.7266	0.8284	0.8318	0.8301
iphone_07_31-3-19-3	0.7511	0.8563	0.8458	0.8510
iphone_07_31-3-21-3	0.7792	0.8791	0.8576	0.8682
iphone_07_31-3-22-3	0.6746	0.7797	0.7986	0.7890
iphone_07_31-3-23-3	0.7048	0.7226	0.9480	0.8201
iphone_07_31-3-24-3	0.6893	0.7573	0.8568	0.8040
iphone_07_31-3-28-3	0.8791	0.9682	0.9042	0.9351
iphone_07_31-3-29-3	0.7971	0.8995	0.8642	0.8815
Min	0.6675	0.6932	0.7608	0.7890
Max	0.9854	0.9981	1.0000	0.9927
Avg	0.8370	0.8625	0.9638	0.9069
Std Dev	0.0860	0.0863	0.0556	0.0532

G.2 ##iphone, SAME SESSION, DIFFERENT ANNOTATOR

File	Accuracy	Precision	Recall	F-score
iphone_07_17-1-2	0.9549	0.9549	1.0000	0.9769
iphone_07_17-1-3	0.9592	0.9592	1.0000	0.9792
iphone_07_17-2-1	0.9854	0.9854	1.0000	0.9927
iphone_07_17-2-3	0.9592	0.9592	1.0000	0.9792
iphone_07_17-3-1	0.9854	0.9854	1.0000	0.9927
iphone_07_17-3-2	0.9549	0.9549	1.0000	0.9769
iphone_07_18-1-2	0.9413	0.9548	0.9848	0.9696
iphone_07_18-1-3	0.8116	0.8135	0.9930	0.8943
iphone_07_18-2-1	0.8915	0.8915	1.0000	0.9426
iphone_07_18-2-3	0.8028	0.8028	1.0000	0.8906
iphone_07_18-3-1	0.8884	0.9147	0.9647	0.9391
iphone_07_18-3-2	0.9134	0.9592	0.9492	0.9542
iphone_07_19-1-2	0.9717	0.9728	0.9987	0.9856
iphone_07_19-1-3	0.8413	0.8417	0.9991	0.9136
iphone_07_19-2-1	0.9140	0.9141	0.9997	0.9550
iphone_07_19-2-3	0.8419	0.8418	0.9998	0.9140
iphone_07_19-3-1	0.8947	0.9159	0.9740	0.9441
iphone_07_19-3-2	0.9487	0.9741	0.9731	0.9736
iphone_07_21-1-2	0.8941	0.9215	0.9657	0.9431
iphone_07_21-1-3	0.8507	0.8672	0.9730	0.9171
iphone_07_21-2-1	0.8369	0.8387	0.9947	0.9101
iphone_07_21-2-3	0.8532	0.8563	0.9937	0.9199
iphone_07_21-3-1	0.8427	0.8477	0.9880	0.9125
iphone_07_21-3-2	0.9053	0.9205	0.9804	0.9495
iphone_07_22-1-2	0.8379	0.8511	0.9798	0.9109
iphone_07_22-1-3	0.7708	0.7734	0.9888	0.8679
iphone_07_22-2-1	0.8018	0.8099	0.9852	0.8890
iphone_07_22-2-3	0.7635	0.7681	0.9878	0.8642
iphone_07_22-3-1	0.8022	0.8302	0.9484	0.8854
iphone_07_22-3-2	0.8255	0.8649	0.9406	0.9012
iphone_07_23-1-2	0.6991	0.7080	0.9642	0.8165
iphone_07_23-1-3	0.7388	0.7372	0.9820	0.8422

Continued. . .

File	Accuracy	Precision	Recall	F-score
iphone_07_23-2-1	0.7324	0.8093	0.8587	0.8333
iphone_07_23-2-3	0.7161	0.7577	0.8820	0.8152
iphone_07_23-3-1	0.7814	0.8164	0.9280	0.8686
iphone_07_23-3-2	0.7140	0.7306	0.9315	0.8189
iphone_07_24-1-2	0.7273	0.7473	0.9380	0.8318
iphone_07_24-1-3	0.7479	0.7722	0.9374	0.8468
iphone_07_24-2-1	0.7311	0.7588	0.9227	0.8328
iphone_07_24-2-3	0.7453	0.7771	0.9219	0.8433
iphone_07_24-3-1	0.7359	0.7474	0.9606	0.8407
iphone_07_24-3-2	0.7349	0.7434	0.9640	0.8395
iphone_07_28-1-2	0.9421	0.9420	1.0000	0.9701
iphone_07_28-1-3	0.9649	0.9649	1.0000	0.9821
iphone_07_28-2-1	0.9592	0.9598	0.9993	0.9791
iphone_07_28-2-3	0.9649	0.9656	0.9993	0.9821
iphone_07_28-3-1	0.9599	0.9599	1.0000	0.9795
iphone_07_28-3-2	0.9428	0.9427	1.0000	0.9705
iphone_07_29-1-2	0.9254	0.9272	0.9978	0.9612
iphone_07_29-1-3	0.8753	0.8754	0.9994	0.9333
iphone_07_29-2-1	0.9377	0.9394	0.9978	0.9677
iphone_07_29-2-3	0.8758	0.8762	0.9988	0.9335
iphone_07_29-3-1	0.9234	0.9469	0.9727	0.9596
iphone_07_29-3-2	0.9136	0.9363	0.9730	0.9543
iphone_07_31-1-2	0.8206	0.8550	0.9388	0.8949
iphone_07_31-1-3	0.8239	0.8457	0.9519	0.8957
iphone_07_31-2-1	0.8140	0.8422	0.9464	0.8913
iphone_07_31-2-3	0.8455	0.8532	0.9728	0.9091
iphone_07_31-3-1	0.8306	0.8634	0.9381	0.8992
iphone_07_31-3-2	0.8654	0.8880	0.9551	0.9204
Min	0.6991	0.7080	0.8587	0.8152
Max	0.9854	0.9854	1.0000	0.9927
Avg	0.8572	0.8706	0.9732	0.9176

Continued. . .

File	Accuracy	Precision	Recall	F-score
Std Dev	0.0840	0.0788	0.0311	0.0530

G.3 ##physics, SAME ANNOTATOR, DIFFERENT SESSION

File	Accuracy	Precision	Recall	F-score
physics_07_17-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_17-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_17-1-21-1	0.9518	0.9518	1.0000	0.9753
physics_07_17-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_17-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_17-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_17-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_17-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_17-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_17-2-18-2	0.9475	0.9948	0.9522	0.9731
physics_07_17-2-19-2	0.8967	0.9091	0.9840	0.9451
physics_07_17-2-21-2	0.9286	0.9691	0.9565	0.9628
physics_07_17-2-22-2	0.9666	0.9817	0.9843	0.9830
physics_07_17-2-23-2	0.9522	0.9755	0.9755	0.9755
physics_07_17-2-24-2	0.7561	0.8242	0.8926	0.8570
physics_07_17-2-28-2	0.6334	0.6538	0.9318	0.7684
physics_07_17-2-29-2	0.8527	0.9204	0.9174	0.9189
physics_07_17-2-31-2	0.9566	0.9641	0.9918	0.9778
physics_07_17-3-18-3	1.0000	1.0000	1.0000	1.0000
physics_07_17-3-19-3	0.9393	0.9393	1.0000	0.9687
physics_07_17-3-21-3	0.9409	0.9409	1.0000	0.9696
physics_07_17-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_17-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_17-3-24-3	0.8645	0.8645	1.0000	0.9273
physics_07_17-3-28-3	0.6845	0.6845	1.0000	0.8127

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_17-3-29-3	0.8387	0.8387	1.0000	0.9123
physics_07_17-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_18-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_18-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_18-1-21-1	0.9518	0.9518	1.0000	0.9753
physics_07_18-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_18-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_18-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_18-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_18-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_18-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_18-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_18-2-19-2	0.9031	0.9031	1.0000	0.9491
physics_07_18-2-21-2	0.9651	0.9651	1.0000	0.9822
physics_07_18-2-22-2	0.9820	0.9820	1.0000	0.9909
physics_07_18-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_18-2-24-2	0.8188	0.8188	1.0000	0.9004
physics_07_18-2-28-2	0.6528	0.6528	1.0000	0.7899
physics_07_18-2-29-2	0.9093	0.9093	1.0000	0.9525
physics_07_18-2-31-2	0.9625	0.9625	1.0000	0.9809
physics_07_18-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_18-3-19-3	0.9393	0.9393	1.0000	0.9687
physics_07_18-3-21-3	0.9409	0.9409	1.0000	0.9696
physics_07_18-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_18-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_18-3-24-3	0.8645	0.8645	1.0000	0.9273
physics_07_18-3-28-3	0.6845	0.6845	1.0000	0.8127
physics_07_18-3-29-3	0.8387	0.8387	1.0000	0.9123
physics_07_18-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_19-1-17-1	0.9688	0.9873	0.9810	0.9841
physics_07_19-1-18-1	0.9656	0.9949	0.9703	0.9825
physics_07_19-1-21-1	0.9415	0.9520	0.9883	0.9699

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_19-1-22-1	0.9961	1.0000	0.9961	0.9981
physics_07_19-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_19-1-24-1	0.8829	0.8937	0.9863	0.9377
physics_07_19-1-28-1	0.8942	0.9130	0.9766	0.9437
physics_07_19-1-29-1	0.9762	0.9871	0.9888	0.9880
physics_07_19-1-31-1	0.9744	0.9743	1.0000	0.9870
physics_07_19-2-17-2	0.9625	0.9618	1.0000	0.9805
physics_07_19-2-18-2	0.9525	0.9949	0.9572	0.9757
physics_07_19-2-21-2	0.9417	0.9691	0.9705	0.9698
physics_07_19-2-22-2	0.9756	0.9819	0.9935	0.9877
physics_07_19-2-23-2	0.9669	0.9759	0.9906	0.9832
physics_07_19-2-24-2	0.7868	0.8346	0.9225	0.8763
physics_07_19-2-28-2	0.6460	0.6646	0.9240	0.7731
physics_07_19-2-29-2	0.8838	0.9204	0.9548	0.9373
physics_07_19-2-31-2	0.9684	0.9683	1.0000	0.9839
physics_07_19-3-17-3	0.9813	1.0000	0.9813	0.9905
physics_07_19-3-18-3	0.9869	1.0000	0.9869	0.9934
physics_07_19-3-21-3	0.9322	0.9408	0.9904	0.9649
physics_07_19-3-22-3	0.9974	1.0000	0.9974	0.9987
physics_07_19-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_19-3-24-3	0.8578	0.8648	0.9904	0.9233
physics_07_19-3-28-3	0.6839	0.6880	0.9847	0.8101
physics_07_19-3-29-3	0.8370	0.8412	0.9931	0.9109
physics_07_19-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_21-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_21-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_21-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_21-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_21-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_21-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_21-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_21-1-29-1	0.9872	0.9872	1.0000	0.9936

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_21-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_21-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_21-2-18-2	0.9934	0.9951	0.9984	0.9967
physics_07_21-2-19-2	0.9043	0.9042	1.0000	0.9497
physics_07_21-2-22-2	0.9820	0.9820	1.0000	0.9909
physics_07_21-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_21-2-24-2	0.8182	0.8187	0.9993	0.9000
physics_07_21-2-28-2	0.6534	0.6533	0.9995	0.7902
physics_07_21-2-29-2	0.9093	0.9095	0.9997	0.9525
physics_07_21-2-31-2	0.9625	0.9625	1.0000	0.9809
physics_07_21-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_21-3-18-3	0.9902	1.0000	0.9902	0.9951
physics_07_21-3-19-3	0.9385	0.9392	0.9991	0.9683
physics_07_21-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_21-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_21-3-24-3	0.8634	0.8644	0.9986	0.9267
physics_07_21-3-28-3	0.6869	0.6861	1.0000	0.8139
physics_07_21-3-29-3	0.8375	0.8390	0.9976	0.9115
physics_07_21-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_22-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_22-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_22-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_22-1-21-1	0.9518	0.9518	1.0000	0.9753
physics_07_22-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_22-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_22-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_22-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_22-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_22-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_22-2-18-2	0.9951	0.9951	1.0000	0.9975
physics_07_22-2-19-2	0.9027	0.9031	0.9996	0.9489
physics_07_22-2-21-2	0.9635	0.9652	0.9981	0.9814

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_22-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_22-2-24-2	0.8154	0.8194	0.9935	0.8981
physics_07_22-2-28-2	0.6528	0.6533	0.9977	0.7896
physics_07_22-2-29-2	0.9056	0.9098	0.9948	0.9504
physics_07_22-2-31-2	0.9625	0.9625	1.0000	0.9809
physics_07_22-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_22-3-18-3	1.0000	1.0000	1.0000	1.0000
physics_07_22-3-19-3	0.9393	0.9393	1.0000	0.9687
physics_07_22-3-21-3	0.9409	0.9409	1.0000	0.9696
physics_07_22-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_22-3-24-3	0.8645	0.8645	1.0000	0.9273
physics_07_22-3-28-3	0.6845	0.6845	1.0000	0.8127
physics_07_22-3-29-3	0.8387	0.8387	1.0000	0.9123
physics_07_22-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_23-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_23-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_23-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_23-1-21-1	0.9518	0.9518	1.0000	0.9753
physics_07_23-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_23-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_23-1-28-1	0.9083	0.9083	1.0000	0.9519
physics_07_23-1-29-1	0.9872	0.9872	1.0000	0.9936
physics_07_23-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_23-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_23-2-18-2	0.9967	0.9967	1.0000	0.9984
physics_07_23-2-19-2	0.9039	0.9038	1.0000	0.9495
physics_07_23-2-21-2	0.9645	0.9651	0.9994	0.9819
physics_07_23-2-22-2	0.9820	0.9820	1.0000	0.9909
physics_07_23-2-24-2	0.8188	0.8188	1.0000	0.9004
physics_07_23-2-28-2	0.6549	0.6542	1.0000	0.7910
physics_07_23-2-29-2	0.9078	0.9092	0.9983	0.9517
physics_07_23-2-31-2	0.9606	0.9625	0.9980	0.9799

Continued…

File	Accuracy	Precision	Recall	F-score
physics_07_23-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_23-3-18-3	1.0000	1.0000	1.0000	1.0000
physics_07_23-3-19-3	0.9405	0.9404	1.0000	0.9693
physics_07_23-3-21-3	0.9405	0.9409	0.9996	0.9693
physics_07_23-3-22-3	1.0000	1.0000	1.0000	1.0000
physics_07_23-3-24-3	0.8645	0.8645	1.0000	0.9273
physics_07_23-3-28-3	0.6869	0.6861	1.0000	0.8139
physics_07_23-3-29-3	0.8375	0.8385	0.9985	0.9115
physics_07_23-3-31-3	0.9270	0.9270	1.0000	0.9621
physics_07_24-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_24-1-18-1	0.9836	0.9967	0.9868	0.9917
physics_07_24-1-19-1	0.9240	0.9243	0.9996	0.9604
physics_07_24-1-21-1	0.9502	0.9525	0.9975	0.9744
physics_07_24-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_24-1-23-1	0.9890	1.0000	0.9890	0.9945
physics_07_24-1-28-1	0.9113	0.9135	0.9967	0.9533
physics_07_24-1-29-1	0.9825	0.9874	0.9949	0.9912
physics_07_24-1-31-1	0.9862	0.9899	0.9959	0.9929
physics_07_24-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_24-2-18-2	0.9361	0.9982	0.9374	0.9669
physics_07_24-2-19-2	0.8995	0.9094	0.9871	0.9466
physics_07_24-2-21-2	0.9454	0.9669	0.9768	0.9718
physics_07_24-2-22-2	0.9782	0.9820	0.9961	0.9890
physics_07_24-2-23-2	0.9835	0.9869	0.9962	0.9916
physics_07_24-2-28-2	0.6791	0.6734	0.9872	0.8007
physics_07_24-2-29-2	0.8828	0.9113	0.9650	0.9374
physics_07_24-2-31-2	0.9546	0.9774	0.9754	0.9764
physics_07_24-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_24-3-18-3	0.9951	1.0000	0.9951	0.9975
physics_07_24-3-19-3	0.9393	0.9407	0.9983	0.9686
physics_07_24-3-21-3	0.9379	0.9413	0.9961	0.9679
physics_07_24-3-22-3	1.0000	1.0000	1.0000	1.0000

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_24-3-23-3	0.9853	0.9870	0.9981	0.9925
physics_07_24-3-28-3	0.6947	0.6916	0.9996	0.8176
physics_07_24-3-29-3	0.8342	0.8382	0.9943	0.9096
physics_07_24-3-31-3	0.9290	0.9340	0.9936	0.9629
physics_07_28-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_28-1-18-1	0.9967	0.9984	0.9984	0.9984
physics_07_28-1-19-1	0.9228	0.9245	0.9978	0.9598
physics_07_28-1-21-1	0.9492	0.9526	0.9962	0.9739
physics_07_28-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_28-1-23-1	0.9871	1.0000	0.9871	0.9935
physics_07_28-1-24-1	0.8953	0.8952	0.9998	0.9446
physics_07_28-1-29-1	0.9815	0.9874	0.9939	0.9906
physics_07_28-1-31-1	0.9665	0.9741	0.9919	0.9829
physics_07_28-2-17-2	0.7938	0.9683	0.8079	0.8809
physics_07_28-2-18-2	0.7590	1.0000	0.7578	0.8622
physics_07_28-2-19-2	0.7945	0.9263	0.8393	0.8806
physics_07_28-2-21-2	0.8056	0.9809	0.8145	0.8900
physics_07_28-2-22-2	0.8511	0.9880	0.8588	0.9189
physics_07_28-2-23-2	0.8676	0.9935	0.8701	0.9277
physics_07_28-2-24-2	0.7049	0.8772	0.7437	0.8050
physics_07_28-2-29-2	0.7483	0.9238	0.7882	0.8506
physics_07_28-2-31-2	0.8876	0.9865	0.8955	0.9388
physics_07_28-3-17-3	0.9438	1.0000	0.9438	0.9711
physics_07_28-3-18-3	0.8525	1.0000	0.8525	0.9204
physics_07_28-3-19-3	0.8979	0.9594	0.9307	0.9448
physics_07_28-3-21-3	0.8800	0.9498	0.9211	0.9353
physics_07_28-3-22-3	0.9435	1.0000	0.9435	0.9709
physics_07_28-3-23-3	0.9577	0.9885	0.9680	0.9782
physics_07_28-3-24-3	0.7622	0.8852	0.8329	0.8582
physics_07_28-3-29-3	0.8094	0.8664	0.9137	0.8894
physics_07_28-3-31-3	0.9231	0.9518	0.9660	0.9588
physics_07_29-1-17-1	0.9875	0.9875	1.0000	0.9937

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_29-1-18-1	0.9951	0.9951	1.0000	0.9975
physics_07_29-1-19-1	0.9228	0.9228	1.0000	0.9598
physics_07_29-1-21-1	0.9506	0.9517	0.9987	0.9747
physics_07_29-1-22-1	1.0000	1.0000	1.0000	1.0000
physics_07_29-1-23-1	1.0000	1.0000	1.0000	1.0000
physics_07_29-1-24-1	0.8936	0.8936	1.0000	0.9438
physics_07_29-1-28-1	0.9071	0.9082	0.9987	0.9513
physics_07_29-1-31-1	0.9724	0.9724	1.0000	0.9860
physics_07_29-2-17-2	0.9438	0.9438	1.0000	0.9711
physics_07_29-2-18-2	0.9951	0.9951	1.0000	0.9975
physics_07_29-2-19-2	0.9039	0.9041	0.9996	0.9495
physics_07_29-2-21-2	0.9643	0.9662	0.9979	0.9818
physics_07_29-2-22-2	0.9807	0.9820	0.9987	0.9903
physics_07_29-2-23-2	0.9743	0.9761	0.9981	0.9870
physics_07_29-2-24-2	0.8161	0.8214	0.9910	0.8982
physics_07_29-2-28-2	0.6552	0.6572	0.9863	0.7888
physics_07_29-2-31-2	0.9625	0.9625	1.0000	0.9809
physics_07_29-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_29-3-18-3	0.9902	1.0000	0.9902	0.9951
physics_07_29-3-19-3	0.9385	0.9439	0.9936	0.9681
physics_07_29-3-21-3	0.9250	0.9414	0.9814	0.9610
physics_07_29-3-22-3	0.9936	1.0000	0.9936	0.9968
physics_07_29-3-23-3	0.9669	0.9777	0.9887	0.9832
physics_07_29-3-24-3	0.8163	0.8690	0.9273	0.8972
physics_07_29-3-28-3	0.6932	0.7055	0.9472	0.8086
physics_07_29-3-31-3	0.9290	0.9289	1.0000	0.9631
physics_07_31-1-17-1	0.9875	0.9875	1.0000	0.9937
physics_07_31-1-18-1	0.9803	0.9950	0.9852	0.9901
physics_07_31-1-19-1	0.9232	0.9232	1.0000	0.9601
physics_07_31-1-21-1	0.9506	0.9519	0.9985	0.9747
physics_07_31-1-22-1	0.9974	1.0000	0.9974	0.9987
physics_07_31-1-23-1	1.0000	1.0000	1.0000	1.0000

Continued...

File	Accuracy	Precision	Recall	F-score
physics_07_31-1-24-1	0.8932	0.8955	0.9968	0.9434
physics_07_31-1-28-1	0.9023	0.9092	0.9914	0.9485
physics_07_31-1-29-1	0.9860	0.9872	0.9987	0.9929
physics_07_31-2-17-2	0.9500	0.9497	1.0000	0.9742
physics_07_31-2-18-2	0.9672	0.9949	0.9720	0.9833
physics_07_31-2-19-2	0.9067	0.9093	0.9960	0.9507
physics_07_31-2-21-2	0.9583	0.9656	0.9921	0.9787
physics_07_31-2-22-2	0.9769	0.9819	0.9948	0.9883
physics_07_31-2-23-2	0.9761	0.9761	1.0000	0.9879
physics_07_31-2-24-2	0.8075	0.8215	0.9773	0.8926
physics_07_31-2-28-2	0.6415	0.6548	0.9533	0.7764
physics_07_31-2-29-2	0.9063	0.9130	0.9915	0.9506
physics_07_31-3-17-3	1.0000	1.0000	1.0000	1.0000
physics_07_31-3-18-3	0.9689	1.0000	0.9689	0.9842
physics_07_31-3-19-3	0.9341	0.9480	0.9837	0.9655
physics_07_31-3-21-3	0.9328	0.9442	0.9869	0.9651
physics_07_31-3-22-3	0.9936	1.0000	0.9936	0.9968
physics_07_31-3-23-3	0.9779	0.9779	1.0000	0.9888
physics_07_31-3-24-3	0.8323	0.8673	0.9516	0.9075
physics_07_31-3-28-3	0.6764	0.6953	0.9385	0.7988
physics_07_31-3-29-3	0.8332	0.8427	0.9851	0.9083
Min	0.6334	0.6528	0.7437	0.7684
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.9200	0.9324	0.9848	0.9554
Std Dev	0.0885	0.0840	0.0388	0.0535

G.4 ##physics, SAME SESSION, DIFFERENT ANNOTATOR

File	Accuracy	Precision	Recall	F-score
physics_07_17-1-2	0.9438	0.9438	1.0000	0.9711
physics_07_17-1-3	1.0000	1.0000	1.0000	1.0000
physics_07_17-2-1	0.9688	0.9873	0.9810	0.9841
physics_07_17-2-3	0.9813	1.0000	0.9813	0.9905
physics_07_17-3-1	0.9875	0.9875	1.0000	0.9937
physics_07_17-3-2	0.9438	0.9438	1.0000	0.9711
physics_07_18-1-2	0.9951	0.9951	1.0000	0.9975
physics_07_18-1-3	1.0000	1.0000	1.0000	1.0000
physics_07_18-2-1	0.9951	0.9951	1.0000	0.9975
physics_07_18-2-3	1.0000	1.0000	1.0000	1.0000
physics_07_18-3-1	0.9951	0.9951	1.0000	0.9975
physics_07_18-3-2	0.9951	0.9951	1.0000	0.9975
physics_07_19-1-2	0.9003	0.9064	0.9920	0.9473
physics_07_19-1-3	0.9357	0.9426	0.9919	0.9666
physics_07_19-2-1	0.9148	0.9277	0.9843	0.9552
physics_07_19-2-3	0.9369	0.9474	0.9876	0.9671
physics_07_19-3-1	0.9212	0.9271	0.9926	0.9588
physics_07_19-3-2	0.9095	0.9113	0.9969	0.9522
physics_07_21-1-2	0.9651	0.9651	1.0000	0.9822
physics_07_21-1-3	0.9409	0.9409	1.0000	0.9696
physics_07_21-2-1	0.9518	0.9518	1.0000	0.9753
physics_07_21-2-3	0.9409	0.9409	1.0000	0.9696
physics_07_21-3-1	0.9512	0.9518	0.9994	0.9750
physics_07_21-3-2	0.9645	0.9651	0.9994	0.9819
physics_07_22-1-2	0.9820	0.9820	1.0000	0.9909
physics_07_22-1-3	1.0000	1.0000	1.0000	1.0000
physics_07_22-2-1	0.9961	1.0000	0.9961	0.9981
physics_07_22-2-3	0.9961	1.0000	0.9961	0.9981
physics_07_22-3-1	1.0000	1.0000	1.0000	1.0000
physics_07_22-3-2	0.9820	0.9820	1.0000	0.9909
physics_07_23-1-2	0.9761	0.9761	1.0000	0.9879
physics_07_23-1-3	0.9779	0.9779	1.0000	0.9888

Continued. . .

File	Accuracy	Precision	Recall	F-score
physics_07_23-2-1	0.9963	1.0000	0.9963	0.9982
physics_07_23-2-3	0.9816	0.9815	1.0000	0.9907
physics_07_23-3-1	0.9945	1.0000	0.9945	0.9972
physics_07_23-3-2	0.9816	0.9815	1.0000	0.9907
physics_07_24-1-2	0.8206	0.8210	0.9987	0.9012
physics_07_24-1-3	0.8663	0.8668	0.9988	0.9281
physics_07_24-2-1	0.8770	0.8968	0.9745	0.9340
physics_07_24-2-3	0.8506	0.8682	0.9752	0.9186
physics_07_24-3-1	0.8959	0.8960	0.9995	0.9449
physics_07_24-3-2	0.8208	0.8208	0.9993	0.9013
physics_07_28-1-2	0.6609	0.6582	0.9995	0.7937
physics_07_28-1-3	0.6926	0.6902	0.9996	0.8165
physics_07_28-2-1	0.7266	0.9459	0.7414	0.8313
physics_07_28-2-3	0.7180	0.7826	0.8141	0.7980
physics_07_28-3-1	0.7888	0.9437	0.8161	0.8753
physics_07_28-3-2	0.7072	0.7292	0.8773	0.7964
physics_07_29-1-2	0.9093	0.9093	1.0000	0.9525
physics_07_29-1-3	0.8387	0.8387	1.0000	0.9123
physics_07_29-2-1	0.9772	0.9871	0.9899	0.9885
physics_07_29-2-3	0.8447	0.8452	0.9976	0.9151
physics_07_29-3-1	0.9384	0.9873	0.9498	0.9682
physics_07_29-3-2	0.8991	0.9256	0.9667	0.9457
physics_07_31-1-2	0.9684	0.9701	0.9980	0.9838
physics_07_31-1-3	0.9329	0.9343	0.9979	0.9650
physics_07_31-2-1	0.9822	0.9859	0.9959	0.9909
physics_07_31-2-3	0.9408	0.9418	0.9979	0.9690
physics_07_31-3-1	0.9763	0.9878	0.9878	0.9878
physics_07_31-3-2	0.9744	0.9817	0.9918	0.9867
Min	0.6609	0.6582	0.7414	0.7937
Max	1.0000	1.0000	1.0000	1.0000
Avg	0.9252	0.9369	0.9826	0.9573

Continued…

File	Accuracy	Precision	Recall	F-score
Std Dev	0.0860	0.0772	0.0485	0.0542

G.5 #python, SAME ANNOTATOR, DIFFERENT SESSION

File	Accuracy	Precision	Recall	F-score
python_07_17-1-18-1	0.7304	0.7638	0.8550	0.8069
python_07_17-1-19-1	0.7095	0.7589	0.8207	0.7886
python_07_17-1-21-1	0.7001	0.7614	0.8257	0.7922
python_07_17-1-22-1	0.7061	0.6459	0.8573	0.7367
python_07_17-1-23-1	0.7398	0.7363	0.8687	0.7970
python_07_17-1-24-1	0.6468	0.5816	0.8770	0.6994
python_07_17-1-28-1	0.7188	0.7994	0.7723	0.7856
python_07_17-1-29-1	0.7489	0.7459	0.8722	0.8041
python_07_17-1-31-1	0.7069	0.7258	0.8501	0.7831
python_07_17-2-18-2	0.7416	0.7400	0.9097	0.8161
python_07_17-2-19-2	0.6745	0.6820	0.8699	0.7646
python_07_17-2-21-2	0.7191	0.7368	0.8968	0.8090
python_07_17-2-22-2	0.6957	0.6390	0.8783	0.7398
python_07_17-2-23-2	0.7398	0.7671	0.8631	0.8123
python_07_17-2-24-2	0.6491	0.5678	0.9305	0.7052
python_07_17-2-28-2	0.7290	0.6859	0.8902	0.7748
python_07_17-2-29-2	0.7368	0.7257	0.8952	0.8016
python_07_17-2-31-2	0.7071	0.7112	0.8825	0.7876
python_07_17-3-18-3	0.7366	0.7572	0.8723	0.8107
python_07_17-3-19-3	0.6950	0.6811	0.8652	0.7622
python_07_17-3-21-3	0.7287	0.7183	0.9047	0.8008
python_07_17-3-22-3	0.7228	0.6729	0.8549	0.7531
python_07_17-3-23-3	0.7412	0.7317	0.8750	0.7969
python_07_17-3-24-3	0.6547	0.5895	0.8830	0.7070
python_07_17-3-28-3	0.7497	0.7103	0.8765	0.7847

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_17-3-29-3	0.7475	0.7392	0.8756	0.8017
python_07_17-3-31-3	0.7004	0.6687	0.8966	0.7661
python_07_18-1-17-1	0.7252	0.7860	0.8056	0.7956
python_07_18-1-19-1	0.7026	0.7765	0.7717	0.7741
python_07_18-1-21-1	0.7001	0.7759	0.7972	0.7864
python_07_18-1-22-1	0.7558	0.7126	0.8224	0.7636
python_07_18-1-23-1	0.7727	0.7794	0.8557	0.8158
python_07_18-1-24-1	0.6882	0.6200	0.8639	0.7219
python_07_18-1-28-1	0.7306	0.8324	0.7465	0.7871
python_07_18-1-29-1	0.7901	0.8028	0.8549	0.8280
python_07_18-1-31-1	0.7429	0.7635	0.8501	0.8045
python_07_18-2-17-2	0.7526	0.8468	0.7877	0.8162
python_07_18-2-19-2	0.7018	0.7482	0.7678	0.7579
python_07_18-2-21-2	0.7448	0.8003	0.8199	0.8099
python_07_18-2-22-2	0.7704	0.7527	0.7948	0.7732
python_07_18-2-23-2	0.7531	0.8315	0.7795	0.8046
python_07_18-2-24-2	0.7475	0.6645	0.8893	0.7607
python_07_18-2-28-2	0.7884	0.7795	0.8309	0.8044
python_07_18-2-29-2	0.7858	0.8197	0.8197	0.8197
python_07_18-2-31-2	0.7521	0.7770	0.8376	0.8062
python_07_18-3-17-3	0.7280	0.7918	0.7958	0.7938
python_07_18-3-19-3	0.7159	0.7151	0.8266	0.7668
python_07_18-3-21-3	0.7520	0.7526	0.8767	0.8100
python_07_18-3-22-3	0.7667	0.7356	0.8244	0.7775
python_07_18-3-23-3	0.7751	0.7773	0.8587	0.8160
python_07_18-3-24-3	0.7057	0.6403	0.8585	0.7335
python_07_18-3-28-3	0.7890	0.7639	0.8604	0.8093
python_07_18-3-29-3	0.7871	0.7964	0.8525	0.8235
python_07_18-3-31-3	0.7317	0.7043	0.8786	0.7818
python_07_19-1-17-1	0.7143	0.7782	0.7970	0.7875
python_07_19-1-18-1	0.7341	0.7663	0.8581	0.8096
python_07_19-1-21-1	0.7076	0.7675	0.8288	0.7970

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_19-1-22-1	0.7201	0.6575	0.8690	0.7486
python_07_19-1-23-1	0.7369	0.7342	0.8662	0.7948
python_07_19-1-24-1	0.6603	0.5930	0.8762	0.7073
python_07_19-1-28-1	0.7379	0.8192	0.7791	0.7986
python_07_19-1-29-1	0.7559	0.7549	0.8690	0.8080
python_07_19-1-31-1	0.7068	0.7318	0.8345	0.7798
python_07_19-2-17-2	0.7493	0.8554	0.7707	0.8108
python_07_19-2-18-2	0.7596	0.7958	0.8321	0.8135
python_07_19-2-21-2	0.7359	0.7945	0.8118	0.8031
python_07_19-2-22-2	0.7683	0.7449	0.8054	0.7739
python_07_19-2-23-2	0.7379	0.8171	0.7707	0.7933
python_07_19-2-24-2	0.7306	0.6479	0.8819	0.7470
python_07_19-2-28-2	0.7816	0.7720	0.8273	0.7987
python_07_19-2-29-2	0.7695	0.8042	0.8089	0.8065
python_07_19-2-31-2	0.7414	0.7770	0.8132	0.7947
python_07_19-3-17-3	0.7271	0.8364	0.7275	0.7782
python_07_19-3-18-3	0.7430	0.8288	0.7594	0.7926
python_07_19-3-21-3	0.7699	0.8021	0.8208	0.8114
python_07_19-3-22-3	0.7829	0.7900	0.7641	0.7768
python_07_19-3-23-3	0.7744	0.8255	0.7752	0.7995
python_07_19-3-24-3	0.7386	0.6950	0.7947	0.7415
python_07_19-3-28-3	0.7960	0.8108	0.7932	0.8019
python_07_19-3-29-3	0.7924	0.8499	0.7817	0.8144
python_07_19-3-31-3	0.7362	0.7461	0.7849	0.7651
python_07_21-1-17-1	0.7157	0.7480	0.8625	0.8012
python_07_21-1-18-1	0.7293	0.7491	0.8856	0.8117
python_07_21-1-19-1	0.7056	0.7379	0.8594	0.7940
python_07_21-1-22-1	0.6989	0.6326	0.8878	0.7388
python_07_21-1-23-1	0.7142	0.7027	0.8911	0.7858
python_07_21-1-24-1	0.6452	0.5773	0.9065	0.7054
python_07_21-1-28-1	0.7378	0.7983	0.8122	0.8052
python_07_21-1-29-1	0.7502	0.7400	0.8902	0.8081

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_21-1-31-1	0.7140	0.7218	0.8789	0.7927
python_07_21-2-17-2	0.7483	0.8388	0.7910	0.8142
python_07_21-2-18-2	0.7559	0.7852	0.8433	0.8132
python_07_21-2-19-2	0.6951	0.7343	0.7810	0.7569
python_07_21-2-22-2	0.7567	0.7324	0.7971	0.7634
python_07_21-2-23-2	0.7427	0.8132	0.7862	0.7995
python_07_21-2-24-2	0.7362	0.6502	0.8986	0.7545
python_07_21-2-28-2	0.7782	0.7635	0.8352	0.7977
python_07_21-2-29-2	0.7714	0.8047	0.8124	0.8085
python_07_21-2-31-2	0.7379	0.7673	0.8240	0.7946
python_07_21-3-17-3	0.7209	0.8107	0.7513	0.7799
python_07_21-3-18-3	0.7476	0.8165	0.7863	0.8011
python_07_21-3-19-3	0.7131	0.7382	0.7628	0.7503
python_07_21-3-22-3	0.7861	0.7931	0.7677	0.7802
python_07_21-3-23-3	0.7801	0.8183	0.7984	0.8082
python_07_21-3-24-3	0.7416	0.6941	0.8088	0.7470
python_07_21-3-28-3	0.8082	0.8210	0.8076	0.8142
python_07_21-3-29-3	0.7848	0.8341	0.7873	0.8100
python_07_21-3-31-3	0.7454	0.7393	0.8259	0.7802
python_07_22-1-17-1	0.7214	0.8486	0.7066	0.7711
python_07_22-1-18-1	0.7227	0.8633	0.6879	0.7657
python_07_22-1-19-1	0.6874	0.8580	0.6309	0.7271
python_07_22-1-21-1	0.6918	0.8357	0.6908	0.7564
python_07_22-1-23-1	0.7742	0.8662	0.7287	0.7915
python_07_22-1-24-1	0.7624	0.7371	0.7663	0.7514
python_07_22-1-28-1	0.6960	0.8830	0.6274	0.7336
python_07_22-1-29-1	0.7894	0.8907	0.7336	0.8046
python_07_22-1-31-1	0.7213	0.8216	0.7051	0.7589
python_07_22-2-17-2	0.7015	0.8995	0.6438	0.7505
python_07_22-2-18-2	0.7266	0.8866	0.6493	0.7496
python_07_22-2-19-2	0.6909	0.8493	0.5975	0.7015
python_07_22-2-21-2	0.7160	0.8576	0.6857	0.7621

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_22-2-23-2	0.7141	0.9098	0.6235	0.7399
python_07_22-2-24-2	0.7882	0.7681	0.7600	0.7640
python_07_22-2-28-2	0.7806	0.8473	0.7088	0.7719
python_07_22-2-29-2	0.7517	0.8929	0.6614	0.7599
python_07_22-2-31-2	0.7173	0.8502	0.6562	0.7408
python_07_22-3-17-3	0.7162	0.8332	0.7110	0.7673
python_07_22-3-18-3	0.7337	0.8455	0.7195	0.7774
python_07_22-3-19-3	0.7245	0.7820	0.7104	0.7445
python_07_22-3-21-3	0.7750	0.8245	0.7962	0.8101
python_07_22-3-23-3	0.7850	0.8604	0.7515	0.8023
python_07_22-3-24-3	0.7481	0.7198	0.7631	0.7408
python_07_22-3-28-3	0.7915	0.8238	0.7624	0.7919
python_07_22-3-29-3	0.7894	0.8772	0.7424	0.8042
python_07_22-3-31-3	0.7384	0.7626	0.7577	0.7602
python_07_23-1-17-1	0.7242	0.8136	0.7585	0.7851
python_07_23-1-18-1	0.7445	0.8223	0.7807	0.8010
python_07_23-1-19-1	0.7116	0.8166	0.7263	0.7688
python_07_23-1-21-1	0.6945	0.7965	0.7507	0.7729
python_07_23-1-22-1	0.7803	0.7644	0.7832	0.7737
python_07_23-1-24-1	0.7341	0.6787	0.8209	0.7431
python_07_23-1-28-1	0.7225	0.8693	0.6874	0.7677
python_07_23-1-29-1	0.7901	0.8418	0.7941	0.8173
python_07_23-1-31-1	0.7307	0.7902	0.7721	0.7811
python_07_23-2-17-2	0.7606	0.8163	0.8474	0.8316
python_07_23-2-18-2	0.7568	0.7681	0.8797	0.8201
python_07_23-2-19-2	0.6868	0.7061	0.8303	0.7632
python_07_23-2-21-2	0.7270	0.7598	0.8605	0.8070
python_07_23-2-22-2	0.7422	0.6980	0.8396	0.7623
python_07_23-2-24-2	0.7048	0.6170	0.9115	0.7359
python_07_23-2-28-2	0.7576	0.7273	0.8596	0.7879
python_07_23-2-29-2	0.7634	0.7721	0.8538	0.8109
python_07_23-2-31-2	0.7387	0.7496	0.8640	0.8028

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_23-3-17-3	0.7275	0.8127	0.7613	0.7862
python_07_23-3-18-3	0.7440	0.8133	0.7838	0.7983
python_07_23-3-19-3	0.7200	0.7444	0.7682	0.7561
python_07_23-3-21-3	0.7662	0.7841	0.8447	0.8133
python_07_23-3-22-3	0.7865	0.7875	0.7782	0.7828
python_07_23-3-24-3	0.7330	0.6816	0.8147	0.7422
python_07_23-3-28-3	0.7899	0.7960	0.8016	0.7988
python_07_23-3-29-3	0.7858	0.8315	0.7931	0.8118
python_07_23-3-31-3	0.7419	0.7379	0.8192	0.7765
python_07_24-1-17-1	0.6987	0.8793	0.6332	0.7362
python_07_24-1-18-1	0.6844	0.9042	0.5826	0.7086
python_07_24-1-19-1	0.6483	0.9036	0.5231	0.6626
python_07_24-1-21-1	0.6716	0.8683	0.6198	0.7233
python_07_24-1-22-1	0.7951	0.8837	0.6596	0.7554
python_07_24-1-23-1	0.7624	0.9236	0.6498	0.7629
python_07_24-1-28-1	0.6604	0.9086	0.5459	0.6820
python_07_24-1-29-1	0.7701	0.9376	0.6545	0.7709
python_07_24-1-31-1	0.6926	0.8591	0.6051	0.7100
python_07_24-2-17-2	0.6760	0.9080	0.5957	0.7194
python_07_24-2-18-2	0.7124	0.9123	0.6014	0.7250
python_07_24-2-19-2	0.6663	0.8661	0.5334	0.6602
python_07_24-2-21-2	0.7252	0.9072	0.6524	0.7590
python_07_24-2-22-2	0.7690	0.8863	0.6089	0.7219
python_07_24-2-23-2	0.6952	0.9298	0.5762	0.7115
python_07_24-2-28-2	0.7787	0.9044	0.6456	0.7534
python_07_24-2-29-2	0.7386	0.9296	0.6059	0.7337
python_07_24-2-31-2	0.7051	0.8718	0.6106	0.7182
python_07_24-3-17-3	0.7077	0.8808	0.6427	0.7431
python_07_24-3-18-3	0.7143	0.8893	0.6373	0.7425
python_07_24-3-19-3	0.7175	0.8352	0.6229	0.7136
python_07_24-3-21-3	0.7805	0.8775	0.7390	0.8023
python_07_24-3-22-3	0.7908	0.8802	0.6678	0.7594

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_24-3-23-3	0.7700	0.9019	0.6774	0.7737
python_07_24-3-28-3	0.7896	0.8748	0.6951	0.7747
python_07_24-3-29-3	0.7551	0.8951	0.6567	0.7576
python_07_24-3-31-3	0.7369	0.8132	0.6739	0.7370
python_07_28-1-17-1	0.7072	0.7366	0.8704	0.7979
python_07_28-1-18-1	0.7301	0.7348	0.9236	0.8184
python_07_28-1-19-1	0.7184	0.7287	0.9137	0.8108
python_07_28-1-21-1	0.7145	0.7410	0.9036	0.8142
python_07_28-1-22-1	0.6691	0.6017	0.9169	0.7266
python_07_28-1-23-1	0.7141	0.6898	0.9339	0.7935
python_07_28-1-24-1	0.6076	0.5481	0.9248	0.6883
python_07_28-1-29-1	0.7352	0.7119	0.9273	0.8055
python_07_28-1-31-1	0.7046	0.7019	0.9128	0.7936
python_07_28-2-17-2	0.7455	0.8535	0.7666	0.8077
python_07_28-2-18-2	0.7640	0.8004	0.8333	0.8165
python_07_28-2-19-2	0.6956	0.7460	0.7570	0.7515
python_07_28-2-21-2	0.7538	0.8152	0.8132	0.8142
python_07_28-2-22-2	0.7611	0.7523	0.7675	0.7598
python_07_28-2-23-2	0.7429	0.8306	0.7610	0.7943
python_07_28-2-24-2	0.7598	0.6821	0.8757	0.7669
python_07_28-2-29-2	0.7816	0.8232	0.8054	0.8142
python_07_28-2-31-2	0.7459	0.7834	0.8115	0.7972
python_07_28-3-17-3	0.7223	0.8078	0.7584	0.7824
python_07_28-3-18-3	0.7487	0.8065	0.8042	0.8053
python_07_28-3-19-3	0.7118	0.7285	0.7812	0.7539
python_07_28-3-21-3	0.7777	0.7920	0.8562	0.8228
python_07_28-3-22-3	0.7752	0.7730	0.7720	0.7725
python_07_28-3-23-3	0.7763	0.8022	0.8158	0.8089
python_07_28-3-24-3	0.7272	0.6740	0.8169	0.7386
python_07_28-3-29-3	0.7854	0.8213	0.8074	0.8143
python_07_28-3-31-3	0.7521	0.7397	0.8440	0.7884
python_07_29-1-17-1	0.7167	0.7927	0.7764	0.7845

Continued…

File	Accuracy	Precision	Recall	F-score
python_07_29-1-18-1	0.7490	0.8065	0.8142	0.8103
python_07_29-1-19-1	0.7037	0.7858	0.7578	0.7715
python_07_29-1-21-1	0.7008	0.7865	0.7797	0.7831
python_07_29-1-22-1	0.7616	0.7277	0.8036	0.7638
python_07_29-1-23-1	0.7731	0.7904	0.8358	0.8125
python_07_29-1-24-1	0.7143	0.6484	0.8523	0.7365
python_07_29-1-28-1	0.7304	0.8502	0.7233	0.7817
python_07_29-1-31-1	0.7397	0.7701	0.8292	0.7985
python_07_29-2-17-2	0.7465	0.8399	0.7863	0.8122
python_07_29-2-18-2	0.7615	0.7900	0.8465	0.8173
python_07_29-2-19-2	0.6964	0.7360	0.7806	0.7576
python_07_29-2-21-2	0.7399	0.7904	0.8273	0.8084
python_07_29-2-22-2	0.7593	0.7341	0.8015	0.7663
python_07_29-2-23-2	0.7442	0.8163	0.7842	0.8000
python_07_29-2-24-2	0.7411	0.6545	0.9025	0.7587
python_07_29-2-28-2	0.7885	0.7722	0.8455	0.8072
python_07_29-2-31-2	0.7466	0.7674	0.8442	0.8039
python_07_29-3-17-3	0.7148	0.7901	0.7714	0.7806
python_07_29-3-18-3	0.7390	0.7882	0.8153	0.8015
python_07_29-3-19-3	0.7084	0.7145	0.8061	0.7575
python_07_29-3-21-3	0.7557	0.7578	0.8740	0.8118
python_07_29-3-22-3	0.7626	0.7391	0.8036	0.7700
python_07_29-3-23-3	0.7688	0.7813	0.8357	0.8076
python_07_29-3-24-3	0.7032	0.6401	0.8474	0.7293
python_07_29-3-28-3	0.7909	0.7704	0.8522	0.8092
python_07_29-3-31-3	0.7255	0.7095	0.8437	0.7708
python_07_31-1-17-1	0.7200	0.7766	0.8120	0.7939
python_07_31-1-18-1	0.7535	0.8021	0.8308	0.8162
python_07_31-1-19-1	0.6950	0.7769	0.7548	0.7657
python_07_31-1-21-1	0.7004	0.7786	0.7927	0.7856
python_07_31-1-22-1	0.7567	0.7164	0.8152	0.7627
python_07_31-1-23-1	0.7706	0.7814	0.8468	0.8128

Continued...

File	Accuracy	Precision	Recall	F-score
python_07_31-1-24-1	0.6924	0.6274	0.8452	0.7202
python_07_31-1-28-1	0.7232	0.8277	0.7390	0.7808
python_07_31-1-29-1	0.7801	0.8012	0.8351	0.8178
python_07_31-2-17-2	0.7488	0.8356	0.7965	0.8156
python_07_31-2-18-2	0.7588	0.7988	0.8251	0.8118
python_07_31-2-19-2	0.6929	0.7480	0.7459	0.7470
python_07_31-2-21-2	0.7359	0.7949	0.8111	0.8029
python_07_31-2-22-2	0.7592	0.7441	0.7788	0.7611
python_07_31-2-23-2	0.7490	0.8313	0.7717	0.8004
python_07_31-2-24-2	0.7328	0.6532	0.8695	0.7460
python_07_31-2-28-2	0.7647	0.7538	0.8177	0.7845
python_07_31-2-29-2	0.7720	0.8105	0.8041	0.8073
python_07_31-3-17-3	0.7114	0.8246	0.7132	0.7648
python_07_31-3-18-3	0.7438	0.8283	0.7616	0.7936
python_07_31-3-19-3	0.7148	0.7535	0.7363	0.7448
python_07_31-3-21-3	0.7839	0.8218	0.8191	0.8204
python_07_31-3-22-3	0.7779	0.8060	0.7253	0.7635
python_07_31-3-23-3	0.7850	0.8356	0.7839	0.8089
python_07_31-3-24-3	0.7475	0.7128	0.7787	0.7443
python_07_31-3-28-3	0.8024	0.8349	0.7731	0.8028
python_07_31-3-29-3	0.7640	0.8343	0.7424	0.7857
Min	0.6076	0.5481	0.5231	0.6602
Max	0.8082	0.9376	0.9339	0.8316
Avg	0.7373	0.7784	0.7919	0.7780
Std Dev	0.0342	0.0741	0.0824	0.0327

G.6 #python, SAME SESSION, DIFFERENT ANNOTATOR

File	Accuracy	Precision	Recall	F-score
python_07_17-1-2	0.7729	0.8322	0.8446	0.8384
python_07_17-1-3	0.7327	0.7761	0.8347	0.8043
python_07_17-2-1	0.7247	0.7559	0.8647	0.8066
python_07_17-2-3	0.7299	0.7553	0.8720	0.8095
python_07_17-3-1	0.7308	0.7767	0.8348	0.8047
python_07_17-3-2	0.7611	0.8211	0.8406	0.8307
python_07_18-1-2	0.7641	0.7825	0.8666	0.8224
python_07_18-1-3	0.7482	0.7828	0.8450	0.8127
python_07_18-2-1	0.7545	0.8123	0.8156	0.8140
python_07_18-2-3	0.7498	0.7996	0.8179	0.8087
python_07_18-3-1	0.7549	0.7992	0.8386	0.8184
python_07_18-3-2	0.7643	0.7854	0.8614	0.8216
python_07_19-1-2	0.7170	0.7276	0.8539	0.7857
python_07_19-1-3	0.7121	0.6943	0.8764	0.7748
python_07_19-2-1	0.7305	0.8148	0.7659	0.7896
python_07_19-2-3	0.7305	0.7381	0.8108	0.7727
python_07_19-3-1	0.7142	0.8411	0.6992	0.7636
python_07_19-3-2	0.7146	0.7937	0.7167	0.7532
python_07_21-1-2	0.7354	0.7495	0.9030	0.8191
python_07_21-1-3	0.7218	0.7031	0.9321	0.8016
python_07_21-2-1	0.7076	0.7870	0.7921	0.7895
python_07_21-2-3	0.7609	0.7609	0.8798	0.8160
python_07_21-3-1	0.7068	0.8113	0.7514	0.7802
python_07_21-3-2	0.7566	0.8273	0.8000	0.8134
python_07_22-1-2	0.7914	0.8316	0.7229	0.7734
python_07_22-1-3	0.7930	0.8357	0.7235	0.7756
python_07_22-2-1	0.7922	0.8391	0.7012	0.7640
python_07_22-2-3	0.7916	0.8568	0.6945	0.7672
python_07_22-3-1	0.7865	0.7991	0.7411	0.7690
python_07_22-3-2	0.7894	0.8168	0.7378	0.7753
python_07_23-1-2	0.7561	0.8590	0.7491	0.8003
python_07_23-1-3	0.7919	0.8273	0.8107	0.8189

Continued. . .

File	Accuracy	Precision	Recall	F-score
python_07_23-2-1	0.7661	0.7660	0.8671	0.8134
python_07_23-2-3	0.7701	0.7633	0.8754	0.8155
python_07_23-3-1	0.7873	0.8308	0.8017	0.8160
python_07_23-3-2	0.7532	0.8573	0.7459	0.7977
python_07_24-1-2	0.8033	0.8212	0.7208	0.7677
python_07_24-1-3	0.7689	0.8040	0.6747	0.7337
python_07_24-2-1	0.7896	0.8399	0.6806	0.7519
python_07_24-2-3	0.7673	0.8150	0.6558	0.7267
python_07_24-3-1	0.7840	0.7996	0.7191	0.7572
python_07_24-3-2	0.8024	0.8008	0.7480	0.7735
python_07_28-1-2	0.7278	0.6745	0.9281	0.7812
python_07_28-1-3	0.7301	0.6739	0.9330	0.7826
python_07_28-2-1	0.7279	0.8690	0.6973	0.7737
python_07_28-2-3	0.8096	0.8082	0.8313	0.8196
python_07_28-3-1	0.7251	0.8732	0.6878	0.7695
python_07_28-3-2	0.8079	0.8156	0.8183	0.8169
python_07_29-1-2	0.7869	0.8170	0.8264	0.8217
python_07_29-1-3	0.7956	0.8148	0.8402	0.8273
python_07_29-2-1	0.8011	0.8264	0.8399	0.8331
python_07_29-2-3	0.7958	0.8151	0.8402	0.8274
python_07_29-3-1	0.7899	0.8150	0.8338	0.8243
python_07_29-3-2	0.7793	0.8088	0.8232	0.8159
python_07_31-1-2	0.7578	0.7709	0.8629	0.8143
python_07_31-1-3	0.7377	0.7068	0.8899	0.7878
python_07_31-2-1	0.7540	0.7818	0.8386	0.8092
python_07_31-2-3	0.7392	0.7146	0.8715	0.7853
python_07_31-3-1	0.7346	0.8098	0.7493	0.7784
python_07_31-3-2	0.7443	0.8124	0.7599	0.7853
Min	0.7068	0.6739	0.6558	0.7267
Max	0.8096	0.8732	0.9330	0.8384
Avg	0.7588	0.7950	0.8027	0.7950

Continued. . .

File	Accuracy	Precision	Recall	F-score
Std Dev	0.0296	0.0459	0.0715	0.0259

APPENDIX H:
CHAT TOOLS PYTHON CODE

The following pages comprise the API documentation for the `chat_tools` Python module. This module provides a suite of general purpose utilities for working with several chat file formats. Many functions require that NLTK[1] and/or WordNet[2] be installed on the system. This code was tested on Mac OS X and Linux operating systems and is known to work with Python version 2.5 (but should be compatible with older versions as well). The full source code will be made available as part of the NPS Chat Corpus [3].

[1] http://nltk.sourceforge.net
[2] http://wordnet.princeton.edu/
[3] http://faculty.nps.edu/cmartell/NPSChat.htm

API Documentation

API Documentation

September 1, 2008

Contents

Contents **1**

1 Module chat_tools **2**

1.1 Functions 2
1.2 Class Post 6
1.2.1 Methods 6
1.3 Class Session 7
1.3.1 Methods 7
1.3.2 Properties 7

Index **9**

1 Module chat_tools

chat_tools.py

This module contains a collection of tools for working with chat.

- Author: P. Adams phadams@nps.edu[1]
- Org: Naval Postgraduate School[2]
- Written: 2008-04-06
- Modified: 2008-05-30

1.1 Functions

main()

Main function for chat_tools.

The chat_tools module should not be called directly. If so, print module information and exit.

demo()

Provide a demo of chat_tool features.

time_diff(*post1*, *post2*, *increment*=`'sec'`)

Return time difference between two posts.

Returns time difference between two posts if time code is available in posts. Returns -1 otherwise.

Optional arguments are:

sec return time in seconds (default) min return time in minutes hour return time in hours day return time in day

get_coll_posts(*chatfile*=`None`)

Return posts from Colloquy chat transcript.

Parses passed Colloquy transcript file or prompts user for Colloquy transcript file if none passed. Returns Post object containing posts and session start and session end.

See http://colloquy.info/ for more info on Colloquy Mac OS X client.

[1] mailto:phadams@nps.edu
[2] http://www.nps.edu

get_lin_posts(*chatfile*=`None`)

Return posts from Lin Chat Corpus.

Parses passed Lin Chat Corpus XML file or prompts user for Lin XML file if none passed. Returns Post object containing post.

**Note: Lin corpus does not contain timestamp info, so posts and session start/end times will be empty.

get_tactical_posts(*chatfile*=`None`)

Return post from tactical chat corpus.

Parses chat from tactical chat XML file.

tokenize_msg(*msg*, *lower*=`True`)

Return tokenized chat message.

Given a message string, returns a list containing all the words in the message. By default, converts message to lower case; can be changed by passing False as second argument.

getnicks(*posts*)

Return nicknames from posts.

Given a list of posts (time, nick, message), returns a dictionary with nicknames as key and frequency (count) as value.

sortnicks_byfreq(*nicks*, *direction*=`'forward'`)

Return dictionary of nicknames sorted by frequency.

Given a dictionary of nicknames, returns dictionary sorted by frequency. Default sort order is ascending; change by passing 'reverse' as second argument.

stopwords_byfreq(*posts*, *number*=`50`)

Return a list of frequency-based stopwords generated from posts.

Returns the top n most frequent words in the posts, where n default is 50.

getalltypes(*posts*)

Return type and frequency for all words in post messages.

getalltokens(*posts*)

Return list of all tokens in passed posts.

getdocvector(*posts*)

Return document vector (list) that represents given posts.

Returned vector dimensions represent all the tokenized, alpha-sorted words in message component of the posts. The value of each dimension is the overall document count for the represented word.

savesession(*session*, *filename*=`None`)

Save chat session to file.

Saves chat session to file. If no filename passed, presents save file dialog. File can be loaded with loadsession(filename).

loadsession(*filename*=`None`)

Load pickled chat session.

Loads from passed filename. If no filename, presents choose file dialog.

anonymize(*posts*)

Anonymize posts (not yet implemented).

This function removes user name and nickname information from a set of posts and returns a list containing two items: 1) dictionary of anonymized names to real user names, nicknames; and 2) list of anonymized chat posts.

exportxml(*posts*)

Export posts to XML.

exportxmpp(*posts*)

Export posts to XMPP (not yet implemented).

exportchattrack(*posts*)

Export posts to Chat Track XML file (not yet implemented).

See http://moby.ittc.ku.edu/chattrack for more info on ChatTrack project.

removemsgs(*posts*, *msg_string*)

Remove posts with messages that match given string and return copy.

Case and white-space sensitive. Returns a copy of the original list with posts consisting of msg_string removed.

enumerate_tf(*posts*)

Enumerates posts using the time field.

Useful when posts do not contain timestamp info (as posts from Lin Corpus). A one-up serialization, starting at 1, will be inserted into the time field of passed posts.

tokenize_posts(*posts*)

Tokenize all posts in session.

Tokenizes all posts in a given set of posts and writes tokenized list to post tokenized message attribute. Also calculates freq distributions.

calc_tfidf(*posts*)

Calculate TFIDF weights for each token in posts in given session.

Requires that posts have been tokenized (tokenize_posts()).

make_conn_matrix(*posts*)

Create connectivity matrix of all passed posts.

get_msg_pairs(*matrix*, *threshold*)

Get message pairs from connectivity matrix.

Given a connectivity matrix and a threshold, return message pairs that comprise posts whose connectivity scores exceed the threshold.

construct_thread(*pairs*, *rmi*)

Construct message thread given root message index (rmi).

recover_thread(*matrix*, *rmi*, *threshold*)

Return message thread given matrix, root message ID, and threshold.

evaluate_pairs(*t_actual*, *pairs*, *label*=0)

Return an evaluation of message pairs against an actual thread.

get_results(*matrix*, *thresholds*, *t_actual*, *label*=0)

Get result of actual

compare(*t_actual*, *t_predict*)

Return results from the comparison of actual message thread with predicted message thread.

nick_augment(*posts*)

Augment msg tokens with user nickname.

time_dist_penalize(*matrix*)

Calculate a time-distance penalized matrix of message posts.

Given an input connectivity matrix, penalizes weights by time-distance given the following formula:

connectivity(i,j) = 1/|i-j| * weight(i,j) if i not equal to j = 0, otherwise

Returns results as a matrix.

hyper_augment(*posts*, *levels*=2)

Augment tokenized post with WN hypernyms.

Scans post tokens and for each word found in WN, adds the n-level hypernyms of first word sense found, where n is the number of levels above in the WN hierarchy.

query_wn(*token*)

Query WN for existence of word.

Returns "Yes" if word is in WordNet, "No" otherwise. Requires that WN be installed and functional on system.

make_token_graph(*posts*, *aug*='aug')

Extract tokens from post and create DOT graph.

Extracts tokens from post msg tokens and builds DOT graph.

1.2 Class Post

Store chat post.

Stores chat post with the following attributes:

user the real user name time received time as time tuple (ref time module) time_org the original received time as formatted in post nick nickname of user on post msg the original message msg_token tokenized message as list msg_aug augmented token list freqdist frequency distribution of the post message tokens tfidf tfidf of given token in post

Note: time/time_org fields do not include timezone information. If preservation of timezone is important, it is recommended to convert time to UTC prior to post instantiation.

1.2.1 Methods

__init__(*self*, *time*, *time_orig*, *user*, *nick*, *msg*, *msg_token*=None, *msg_aug*=None, *freqdist*=None, *tfidf*=None)

1.3 Class Session

object
  list
    chat_tools.Session

Store set of class posts.

1.3.1 Methods

__init__(*self*, *posts*=None, *start*=None, *end*=None, *freqdist*=None)

x.__init__(...) initializes x; see x.__class__.__doc__ for signature

Return Value
new list

Overrides: object.__init__ extit(inherited documentation)

duration(*self*, *increment*='sec')

Return duration of chat session.

Returns duration of the session if start and end times are available. Returns -1 otherwise.

Optional arguments are:

sec return time in seconds (default) min return time in minutes hour return time in hours day return time in day

getallnicks(*self*)

Return all nicknames in session.

main()

Inherited from list

__add__(), __contains__(), __delitem__(), __delslice__(), __eq__(), __ge__(), __getattribute__(), __getitem__(), __getslice__(), __gt__(), __hash__(), __iadd__(), __imul__(), __iter__(), __le__(), __len__(), __lt__(), __mul__(), __ne__(), __new__(), __repr__(), __reversed__(), __rmul__(), __setitem__(), __setslice__(), append(), count(), extend(), index(), insert(), pop(), remove(), reverse(), sort()

Inherited from object

__delattr__(), __reduce__(), __reduce_ex__(), __setattr__(), __str__()

1.3.2 Properties

Name	Description
Inherited from object	
__class__	

Index

chat_tools (module), 2–8
- chat_tools.anonymize *(function)*, 4
- chat_tools.calc_tfidf *(function)*, 5
- chat_tools.compare *(function)*, 5
- chat_tools.construct_thread *(function)*, 5
- chat_tools.demo *(function)*, 2
- chat_tools.enumerate_tf *(function)*, 4
- chat_tools.evaluate_pairs *(function)*, 5
- chat_tools.exportchattrack *(function)*, 4
- chat_tools.exportxml *(function)*, 4
- chat_tools.exportxmpp *(function)*, 4
- chat_tools.get_coll_posts *(function)*, 2
- chat_tools.get_lin_posts *(function)*, 2
- chat_tools.get_msg_pairs *(function)*, 5
- chat_tools.get_results *(function)*, 5
- chat_tools.get_tactical_posts *(function)*, 3
- chat_tools.getalltokens *(function)*, 3
- chat_tools.getalltypes *(function)*, 3
- chat_tools.getdocvector *(function)*, 3
- chat_tools.getnicks *(function)*, 3
- chat_tools.hyper_augment *(function)*, 6
- chat_tools.loadsession *(function)*, 4
- chat_tools.main *(function)*, 2
- chat_tools.make_conn_matrix *(function)*, 5
- chat_tools.make_token_graph *(function)*, 6
- chat_tools.nick_augment *(function)*, 5
- chat_tools.Post *(class)*, 6
  - chat_tools.Post.__init__ *(method)*, 6
- chat_tools.query_wn *(function)*, 6
- chat_tools.recover_thread *(function)*, 5
- chat_tools.removemsgs *(function)*, 4
- chat_tools.savesession *(function)*, 4
- chat_tools.Session *(class)*, 6–8
  - chat_tools.Session.duration *(method)*, 7
  - chat_tools.Session.getallnicks *(method)*, 7
  - chat_tools.Session.main *(method)*, 7
- chat_tools.sortnicks_byfreq *(function)*, 3
- chat_tools.stopwords_byfreq *(function)*, 3
- chat_tools.time_diff *(function)*, 2
- chat_tools.time_dist_penalize *(function)*, 5
- chat_tools.tokenize_msg *(function)*, 3
- chat_tools.tokenize_posts *(function)*, 5

Bibliography

[1] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[2] B. Eovito, "Assessment of joint chat requirements from current usage patterns," Master's thesis, Naval Postgraduate School, Monterey, California, June 2006.

[3] R. Cunningham, O. Amezcua, and L. Duffy, "FY08 Tactical Chat Survey," Navy Space and Naval Warfare Command San Diego, Tech. Rep., 2008.

[4] J. Lin, "Automatic author profiling of online chat logs," Master's thesis, Naval Postgraduate School, Monterey, California, March 2007.

[5] E. N. Forsyth, "Improving automated lexical and discourse analysis of online chat dialog," Master's thesis, Naval Postgraduate School, Monterey, California, September 2007.

[6] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," 2000. [Online]. Available: citeseer.ist.psu.edu/stolcke00dialogue.html

[7] T. Wu, F. M. Khan, T. A. Fisher, L. A. Schuler, and W. M. Pottenger, "Posting act tagging using tranformation-based learning," in *Proceedings of the Workshop on Foundations of Data Mining and Discovery, IEEE International Conference on Data Mining*, December 2002.

[8] E. Ivanovic, "Dialogue act tagging for instant messaging chat sessions," in *ACL*, 2005.

[9] M. Zitzen and D. Stein, "Chat and conversation: a case of transmedial stability?" *Linguistics*, vol. 42, no. 5, pp. 983–1021, 2004.

[10] H. Sacks, E. A. Schegloff, and G. Jefferson, "A simplest systematics for the organization of turn-taking for conversation," *Language*, vol. 50, no. 4, pp. 696–735, 1974.

[11] P. M. Aoki, M. H. Szymanski, L. D. Plurkowski, J. D. Thornton, A. Woodruff, and W. Yi, "Where's the "party" in "multi-party"?: analyzing the structure of small-group sociable talk," in *CSCW '06: Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*. New York, NY, USA: ACM, 2006, pp. 393–402.

[12] M. M. Egbert, “Schisming: The transformation from a single conversation to multiple conversations,” Ph.D. dissertation, Univ. of California, Los Angeles, CA, 1993.

[13] V. H. Tuulos and H. Tirri, “Combining topic models and social networks for chat data mining,” in *WI ’04: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 206–213.

[14] J. O’Neill and D. Martin, “Text chat in action,” in *GROUP ’03: Proceedings of the 2003 International ACM SIGGROUP conference on Supporting group work*. New York, NY, USA: ACM, 2003, pp. 40–49.

[15] S. Herring, “Interactional coherence in CMC,” *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on System Sciences*, vol. Track2, 1999.

[16] M. Elsner and E. Charniak, “You talking to me? A corpus and algorithm for conversation disentanglement,” in *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, June 2008, pp. 834–842. [Online]. Available: http://www.aclweb.org/anthology/P/P08/P08-1095

[17] K. Nigam, J. Lafferty, and A. McCallum, “Using maximum entropy for text classification,” *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pp. 61–67, 1999.

[18] M. Elsner. (2008, June) You talking to me? A corpus and algorithm for conversation disentanglement. [Online]. Available: http://www.cs.brown.edu/~melsner/acl-08-slides.pdf

[19] G. Salton, G. Salton, A. Wong, C. S. Yang, and C. S. Yang, “A vector space model for automatic indexing,” *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[20] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2000.

[21] T. Hofmann, “Probabilistic latent semantic analysis,” in *Proc. of Uncertainty in Artificial Intelligence, UAI’99*, Stockholm, 1999. [Online]. Available: citeseer.ist.psu.edu/hofmann99probabilistic.html

[22] D. Blei, A. Ng, and M. Jordan, “Latent Dirichlet Allocation,” *Journal of Machine Learning Research 3*, 2003. [Online]. Available: http://www.cs.princeton.edu/~blei/papers/BleiNgJordan2003.pdf#search=%22Latent%20Dirichlet%20Allocation%22

[23] T. S. Ferguson, “A Bayesian analysis of some nonparametric problems,” *Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973.

[24] D. M. Blei, T. L. Griffiths, M. Jordan, and J. B. Tenenbaum, “Hierarchical topic models and the nested Chinese restaurant process,” *Neural Information Processing Systems (NIPS) 16*, 2003.

[25] Y. Wang, M. Yoshi, W. Cohen, and C. Rosé, “Recovering implicit thread structure in newsgroup style conversations,” in *Proceedings of the 2nd International Conference on Weblogs and Social Media*, 2008. [Online]. Available: http://www.cs.cmu.edu/~yichiaw/

[26] M. Steyvers and T. Griffiths, *Latent Semantic Analysis: A Road to Meaning*, 2006, ch. Probabilistic Topic Models.

[27] Netlingo. [Online]. Available: http://netlingo.com/

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. LorRaine T. Duffy, Ph.D.
   SSC San Diego Code 246207
   San Diego, California

4. Thuy Nguyen
   Naval Postgraduate School
   Monterey, California